Smart Components

Single Pair Ethernet Sensor and Actuator Adapter

Datasheet

Perinet
Seamless IoT Connectivity

Abstract

This technical datasheet describes the Perinet IIoT *Smart Components* intended to be used for connecting analogue as well as digital sensors and actuators to the Local Area Network via Single Pair Ethernet (SPE).

Doc-No.: PRN.100.387 rev: 3

2024-02-08

Document Information

Title	Smart Components
Subtitle	Single Pair Ethernet Sensor and Actuator Adapter
Type	Datasheet
Status	Release
Version	3
Date	2024-02-08
Disclosure Restriction	

Contents

1 Introduction

Perinet Smart Components facilitate direct IP-based communication between individual sensors or actuators and IT systems, such as Manufacturing Execution Systems (MES), IoT platforms, and enterprise IT networks. This integration into digitized business processes offers significant advantages, primarily in system simplification and cost reduction. By using IP protocols across all levels, complex intermediary systems and the need for specialized installation and maintenance expertise are substantially reduced.

Figure 1 illustrates the integration of Perinet Smart Components within an IIoT environment. Sensors and actuators are connectable to a hybrid Single Pair Ethernet segment through a periNODE. A periSWITCH can link multiple sensors to the network, while a periSTART serves as a bridge between the Single Pair Ethernet segment and existing IT infrastructure, also acting as the central power source for the segment. Consequently, Perinet Smart Components enable a direct, streamlined connection of sensors and actuators to IT systems, eliminating the need for additional converters or components.

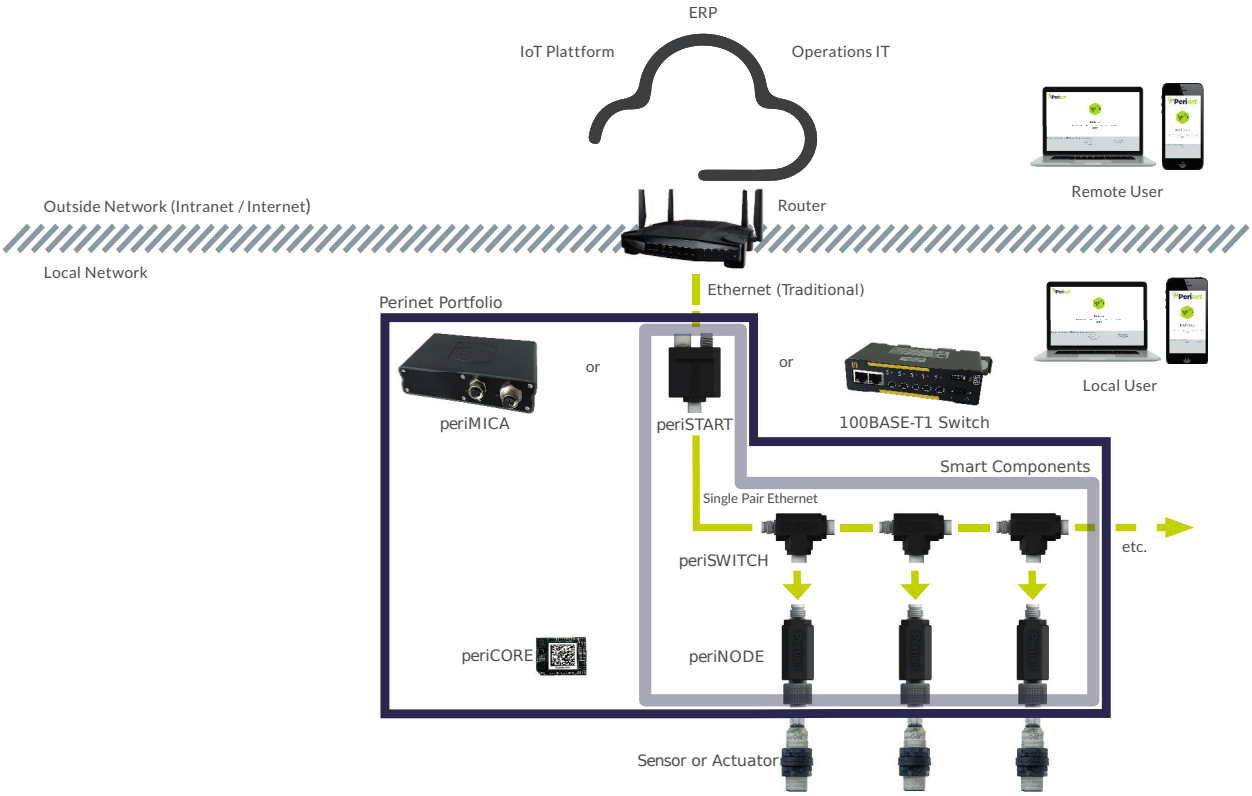

Figure 1: Perinet IP-based communication architecture

A periNODE transforms a passive sensor or actuator into an active, intelligent device. This change means that the sensor now offers information - such as unit-based, compensated, and error-corrected values - rather than just raw data. A key advantage of digital transmission is that data quality remains unaffected by wire length. Additionally, data can be enriched with metadata like timestamps, measurement points, and methods, making it self-descriptive and ideal for integration into higher-level IT systems. Furthermore, sensors can actively send data upon event occurrence instead of relying on constant polling, which can greatly reduce network bandwidth usage.

Ethernet communication capability for individual sensors is a remarkable feature in itself, but Perinet also emphasizes usability in its Smart Components. These components are equipped with a Zeroconf mechanism, enabling their use irrespective of network settings and without any manual configuration. Additionally, Perinet incorporates a state-of-the-art network security system based on certificates, which can be managed automatically for user convenience.

1.1 Product Features

- **IPv6-Based Connectivity**: Supports the latest internet protocol for enhanced network addressability and scalability.
- **Network Protocols**: Utilizes standard IT network protocols like TCP/UDP for reliable data transmission.
- **Security Features**:
  - *End-to-End Encryption (E2EE)*: Ensures secure data transfer using TLS.
  - *Role-Based Access Control (RBAC)*: Manages user permissions efficiently.
  - *Integration with Public Key Infrastructures (PKI)*: Enhances security management.
  - *Confidentiality with TLS 1.2*: Maintains data privacy during transmission.
- **Web Interface**: Includes an HTTPS server hosting a web-GUI for displaying sensor values and a configuration page.
- **RESTful API**: Provides access to sensor values and configuration settings.
- **MQTT[s] Support**: Facilitates publish/subscribe messaging for real-time data updates.
- **Zeroconf Networking**:
  - *DNS Service Discovery*: Simplifies network services' discovery.
  - *mDNS-Based periNODE-Hostname Resolution*: Offers unique hostname resolution within the network.
- **Hybrid Single Pair Ethernet**: A single cable solution for both power supply and data transmission.
- **Extendable Firmware Architecture**: Allows for firmware enhancements and customizations.

2 Perinet Smart Components Portfolio

2.1 periNODE Smart Adapter

The periNODE smart adapter is designed to transform analog or passive digital devices, such as sensors and actuators, into smart, active participants in a network. Its plug & play capability enables easy, quick, and flexible deployment in various settings, including both industrial and commercial environments.

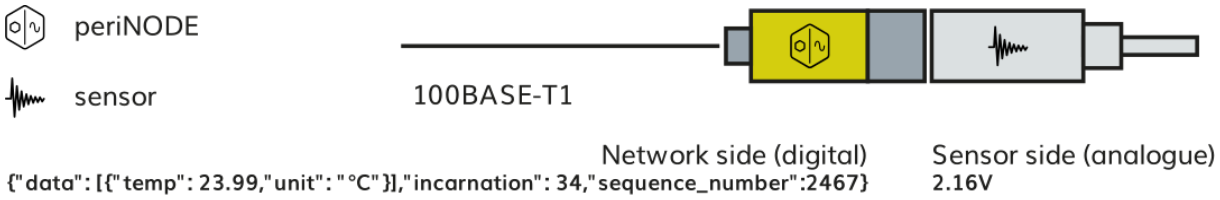

Figure 2: Semantic metrics of the periNODE

A periNODE smart adapter enhances raw sensor signals by converting them into unit-based values and transmits this data, complete with physical units, in a digital format via Ethernet. This allows seamless integration with any IT-system or data hub for further processing and analytics, as illustrated in Figure 2. Additionally, the smart adapter enhances the security of the sensor network by implementing authenticated end-to-end data encryption. The periNODE also offers customizable data communication intervals, enabling users to tailor network traffic and data collection according to their specific requirements.

Note: It is important to note that the *periNODE* smart adapter is not intended for use in hard real-time or safety-critical applications.

Note: For optimal performance, a *periNODE* should be directly connected to the sensor or actuator, or via a cable no longer than 3 meters.

2.1.1 periNODE 0-10V

Figure 3: periNODE 0-10V

This periNODE smart adapter connects directly to sensors with 0-10V signal output and M12 A-coded 4-pin connectors:

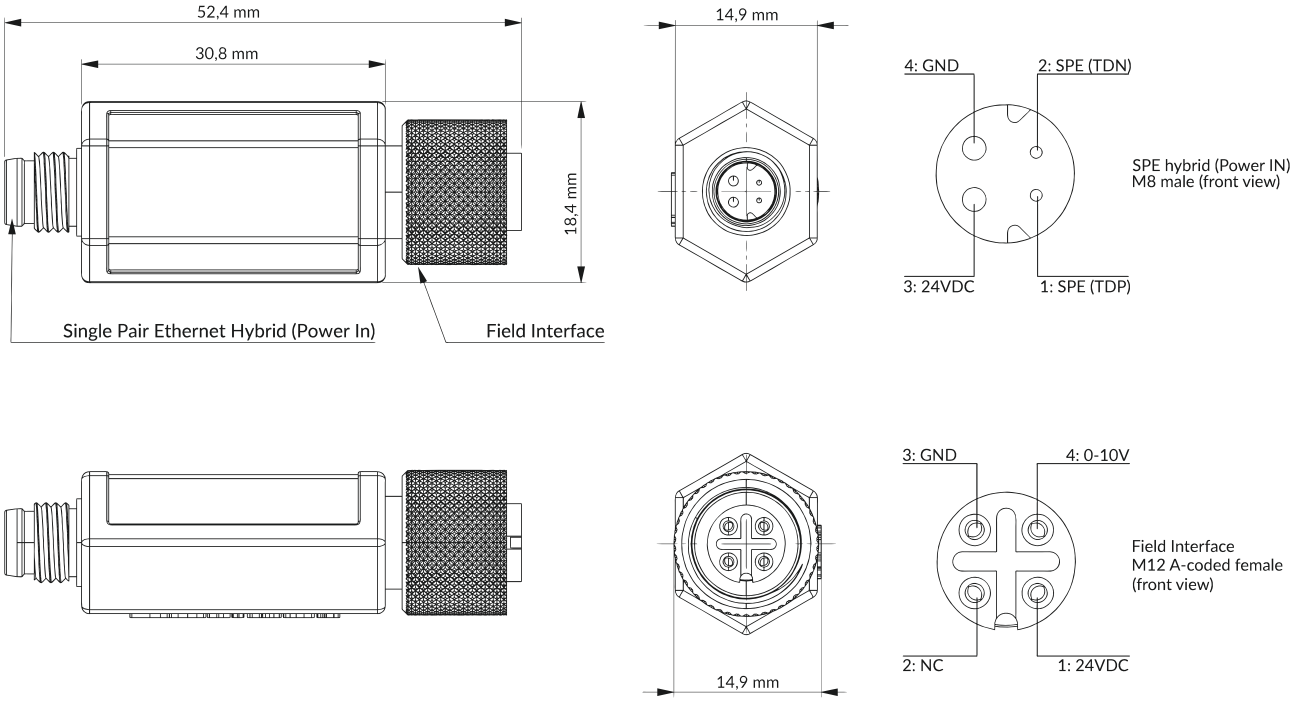

Figure 4: periNODE 0-10V dimensional drawing & pinout

periNODE 0-10V interfaces:

Single Pair Ethernet Hybrid Interface (Network side)	
Connector type	M8 H-coded male connector according to IEC 63171-6:2021 (style 6J-M8CI without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Field Interface (Sensor side)	
Connector type	M12 A-coded 4-pin female connector with free rotating screw nut for direct sensor attachment
Signal	0–10V analogue input
Power	24VDC output, 100mA max

Table 1: periNODE 0-10V interfaces

The standard sampling rate for sensor values in the periNODE smart adapter is set at 1 SPS (samples per second). This rate can be configured up to a maximum of 10 SPS, allowing flexibility to match specific application requirements.

Supported sensors:

- i.a. Micro Detectors ultrasonic sensor UK6A/H1-0EUL

periNODE 0-10V mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	52.4 mm
		Width	18.4 mm
		Height	14.9 mm
		Weight	19 grams
	Material		Thermoplastic material
	Protection		IP67

Table 2: periNODE 0-10V mechanical specifications

periNODE 0-10V electrical specifications:

Electrical Specifications			
Network Interface	Connector	Type	M8 H-coded male
		Standard	Similar to IEC 63171-6:2021 Style 6J-M8CI
		Mating	Mates with periLINE and IEC 63171-6:2021 Style 6P-M8CI
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24V DC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Range	±10%
	Consumption	Average	0.5 W
Sensor Interface	Connector	Kind	Circular connector
		Type	M12 A coded 4pin female
	Signal	Pin 1	24VDC out
		Pin 2	do not connect
		Pin 3	GND
		Pin 4	0–10V input
		Input Current	-10mA – +10mA
		Resolution	<1mV
		Sample Rate	typical: 5 SPS

Table 3: periNODE 0-10V electrical specifications

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V – +40V
100BASE-T1	-40V – +40V
0-10V	-2.7V – +40V

Table 4: periNODE 0-10V absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C – +70°C
Storage conditions	-45°C – +85°C

Table 5: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.1.2 periNODE Pt100

Figure 5: periNODE Pt100

The periNODE smart adapter is designed to be directly compatible with Pt100 temperature sensors. For successful integration, the Pt100 sensor should feature a 3-wire signal output and be equipped with an M12 A-coded 4-pin connector:

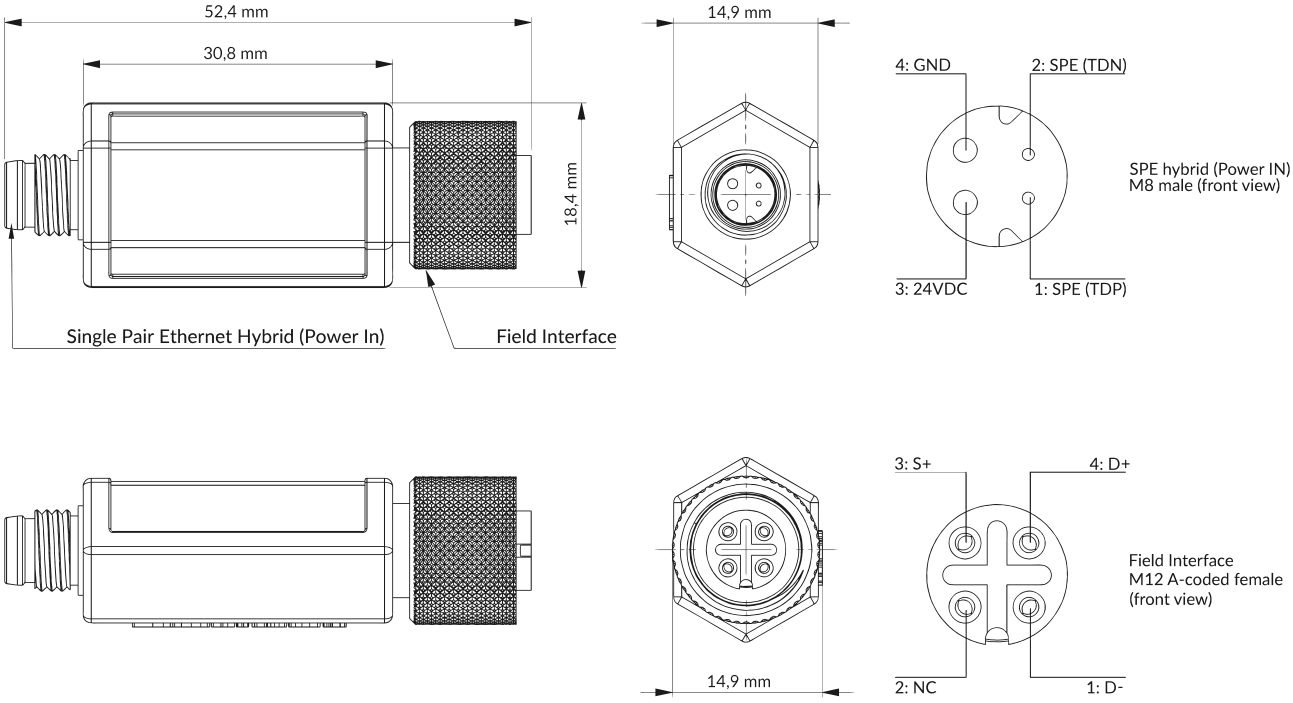

Figure 6: periNODE Pt100 dimensional drawing & pinout

The periNODE Pt100 is specifically engineered to measure temperature using a 3-wire-Pt100 temperature sensor, employing the ratiometric 3-wire RTD (Resistance Temperature Detector) measurement method. This approach is highly resilient against connection failures between the temperature detector (the actual Pt100 resistor) and the analog front end of the periNODE Pt100. Temperature readings are digitized using a Delta-Sigma-ADC (Analog-to-Digital Converter). The raw data collected is then refined through a 4th order parabolic adjustment, resulting in the final, accurate temperature value.

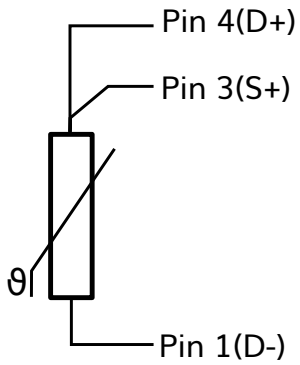

Figure 7: periNODE Pt100 3-wire connection

periNODE Pt100 interfaces:

Single Pair Ethernet Hybrid Interface (Network side)	
Connector type	M8 H-coded male connector according to IEC 63171-6:2021 (style 6J-M8CI without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Field Interface (Sensor side)	
Connector type	M12 A-coded 4-pin female connector with free rotating screw nut for direct sensor attachment
Signal	Pt100 3-wire
Power	24VDC output

Table 6: periNODE Pt100 interfaces

Supported sensors:

- i.a. WIKA miniature resistance thermometer TR33-Z-P3

periNODE Pt100 mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	52.4 mm
		Width	18.4 mm
		Height	14.9 mm
		Weight	19 grams
	Material		Thermoplastic material
	Protection		IP67

Table 7: periNODE Pt100 mechanical specifications

periNODE Pt100 electrical specifications:

Electrical Specifications			
Network Interface	Connector	Type	M8 H-coded male
		Standard	Similar to IEC 63171-6:2021 Style 6J-M8CI
		Mating	Mates with periLINE and IEC 63171-6:2021 Style 6P-M8CI
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24V DC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Range	±10%
	Consumption	Average	0.5 W
Sensor Interface	Connector	Kind	Circular connector
		Type	M12 A coded 4pin female
	Signal	Pin 1	D-
		Pin 2	do not connect

Electrical Specifications			
		Pin 3	S+
		Pin 4	D+
		Current	10µA – 1.5mA
		Resolution	0.035 K
		Sample Rate	typical: 1 SPS (firmware)

Table 8: periNODE Pt100 electrical specifications

The default sample rate for sensor values is 1 SPS (samples per second) and can be configured to a value of maximum 10 SPS.

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V – +40V
100BASE-T1	-40V – +40V
Pt100 Interface	-0.3V – +3.60V

Table 9: periNODE Pt100 absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C – +70°C
Storage conditions	-45°C – +85°C

Table 10: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.1.3 periNODE GPIO

Figure 8: periNODE GPIO

The periNODE smart adapter is equipped with two digital input/output channels operating at a 24V logic level. This makes it compatible for direct connection with sensors that have a digital 24V output or actuators with a digital 24V input. The periNODE GPIO model includes a pigtail cable outlet on the sensor or actuator side (4x0.25mm^2). Users have the flexibility to either equip it with a connector of their choice or connect it to a clamp terminal.

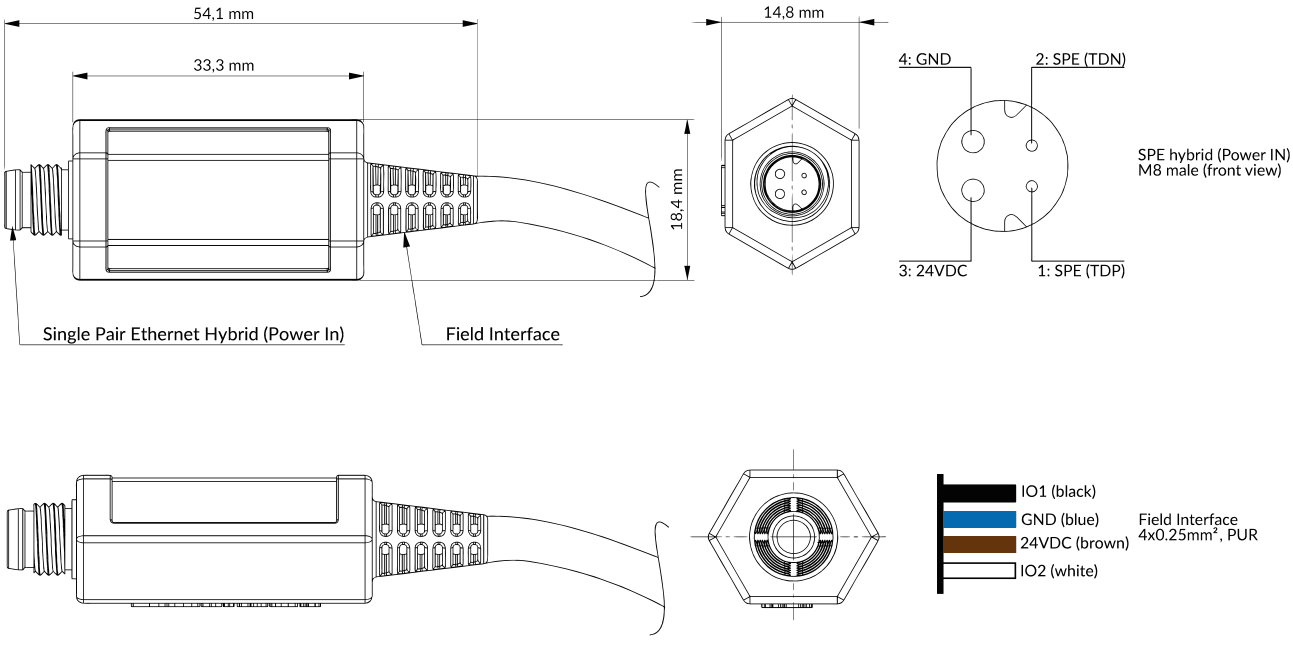

Figure 9: periNODE GPIO dimensional drawing & pinout

periNODE GPIO interfaces:

Single Pair Ethernet Hybrid Interface (Network side)	
Connector type	M8 H-coded male connector according to IEC 63171-6:2021 (style 6J-M8CI without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Field Interface (Sensor or Actuator side)	
Connector type	4x0.25mm², PUR, open-ended cable
Signal	24V digital input or output
Power	24VDC output

Table 11: periNODE GPIO interfaces

The periNODE GPIO is designed to generate events exclusively for ports configured as inputs.

By default, the sampling rate for a GPIO input port is set at 1 SPS (samples per second). Users can adjust this sampling rate up to a maximum of 10 SPS through the configuration interface.

Supported devices:

- various types of switch contacts or proximity switches in industrial and commercial settings

periNODE GPIO mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	54.1 mm + cable ≈ 500 mm
		Width	18.4 mm
		Height	14.8 mm
		Weight	22 grams
	Material		Thermoplastic material
	Protection		IP67

Table 12: periNODE GPIO mechanical specifications

periNODE GPIO electrical specifications:

Electrical Specifications			
Network Interface	Connector	Type	M8 H-coded male
		Standard	Similar to IEC 63171-6:2021 Style 6J-M8CI
		Mating	Mates with periLINE and IEC 63171-6:2021 Style 6P-M8CI
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24V DC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Range	±10%
	Consumption	Average	0.5 W
Sensor or Actuator Interface	Connector	Kind	Pigtail cable
		Type	4x0.25mm², PUR
	Signal	brown	24VDC out
		white	IO2 (default Input)
		blue	GND
		black	IO1 (default Output)
		Input Logic Levels	logic 0 max. 4.3V, logic 1 min. 18.5V
		Input Current	-400µA – +400µA
		Input Type	high impedance
		Output Current	continuous: 100mA, peak <1s: 500mA, reverse current: -90µA – +300µA
		Output Type	Push Pull
		Output rising timing	3.3µs

Electrical Specifications			
		Output falling timing	3.3µs
		Sample Rate	max: 20 SPS (software)
			typical: 0 SPS (event driven)

Table 13: periNODE GPIO electrical specifications

Note: *Caution is advised for each channel configured as* output. *A short circuit to GND (ground) at these pins can potentially damage the device due to excessive heat dissipation.*

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V – +40V
100BASE-T1	-40V – +40V
GPIO	continuous: -36V – +36V, peak <100μs: -52V – +52V

Table 14: periNODE GPIO absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C – +70°C
Storage conditions	-45°C – +85°C

Table 15: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.2 periSTART Media Converter

The periSTART media converter serves as a link between your pre-existing network infrastructure and the Perinet Seamless IoT Connectivity system. It functions as a bridge, converting Fast Ethernet (100BASE-TX) to Single Pair Ethernet (100BASE-T1), while also supplying a 24VDC feed to connected sensors and actuators.

This device combines data and power, transmitting both through the 4-wire periLINE hybrid SPE cable to the IoT Connectivity system.

Note: It is important to note that the *periSTART* media converter is not intended for use in hard real-time or safety-critical applications.

Note: For optimal performance, a *periSTART* media converter should be connected to a traditional ethernet switch and power supply using a cable no longer than 3 meters.

Key Features:

- **Conversion Capability**: Seamlessly converts between 100BASE-TX (Fast Ethernet) and 100BASE-T1 (Single Pair Ethernet) standards.
- **Power Supply**: Delivers a 24VDC power feed to attached devices.
- **Integrated Data and Power Transmission**: Bundles both data and power in a single solution, leading to reduced cabling needs and simplified deployment.

2.2.1 periSTART standard

Figure 10: periSTART standard

The periSTART standard variant is designed for field applications and features M8 connectors, facilitating seamless connectivity with infrastructure components:

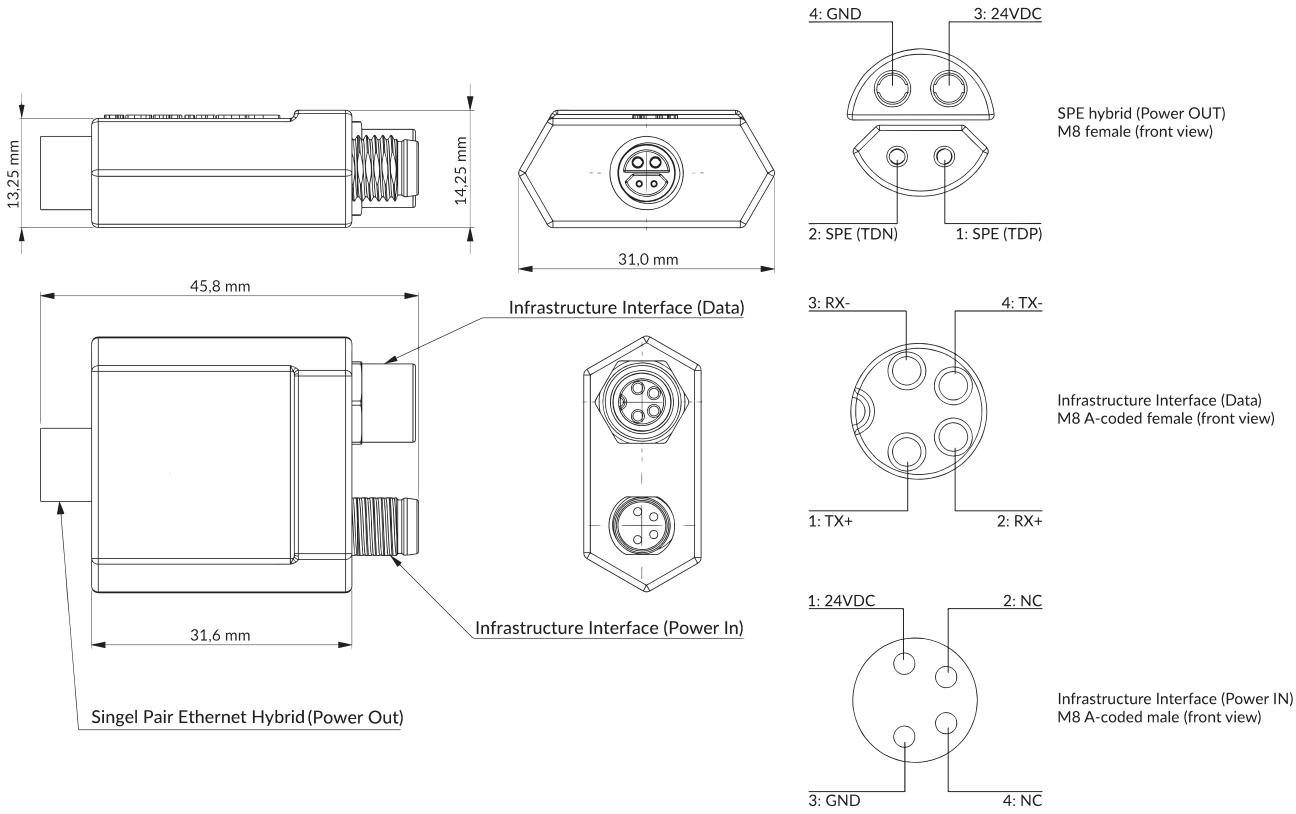

Figure 11: periSTART standard dimensional drawing & pinout

periSTART standard interfaces:

Single Pair Ethernet Hybrid Interface (Power Out)	
Connector type	M8 H-coded female connector according to IEC 63171-6:2021 (style 6J-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC output, continuous 2A max
Infrastructure Interface (Data)	
Connector type	M8 A-coded female connector
Communication	100BASE-TX Fast Ethernet (IEEE 802.3u)
Infrastructure Interface (Power In)	
Connector type	M8 A-coded male connector
Input Voltage	24VDC (±10%)
Consumption	Average 0.5W (max. 2A throughput current)

Table 16: periSTART standard interfaces

periSTART standard mechanical specifications:

Mechanical Specification			
Housing	Dimensions	Length	45.8 mm
		Width	31.0 mm
		Height	14.25 mm
		Weight	29 grams
	Material		Thermoplastic material
	Protection		IP67

Table 17: periSTART standard mechanical specifications

periSTART standard electrical specifications:

Electrical Specifications			
Network Interface 1	Connector	Type	M8 H-coded female
		Standard	Similar to IEC 63171-6:2021 Style 6J-M8C
		Mating	Mates with periLINE and IEC 63171-6:2021 Style 6P-M8C
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24VDC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Direction	Output
Network Interface 2	Connector	Type	M8 A-coded female
		Pin 1	TXP
		Pin 2	RXP
		Pin 3	RXN
		Pin 4	TXN
	Communication	Type	Fast Ethernet
		Physical Layer	100BASE-TX
		Standard	IEEE 802.3u
Power interface	Connector	Type	M8 A-coded male
		Pin 1	24VDC
		Pin 2	NC
		Pin 3	GND
		Pin 4	NC
	Input voltage	Nominal	24VDC
		Range	+/-10%

Electrical Specifications			
	Consumption	Average	0.5 W
	Throughput current	Max.	2A

Table 18: periSTART standard electrical specifications

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V – +40V
100BASE-T1	-40V – +40V
100BASE-TX	-40V – +40V

Table 19: periSTART standard absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C – +70°C
Storage conditions	-45°C – +85°C

Table 20: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

2.3 periSWITCH Multi-Port Switch

2.3.1 periSWITCH 3-port

Figure 12: periSWITCH 3-port

The periSWITCH 3-port switch is adeptly designed to enable the series bidirectional connection of multiple sensors and/or actuators in a network configuration.

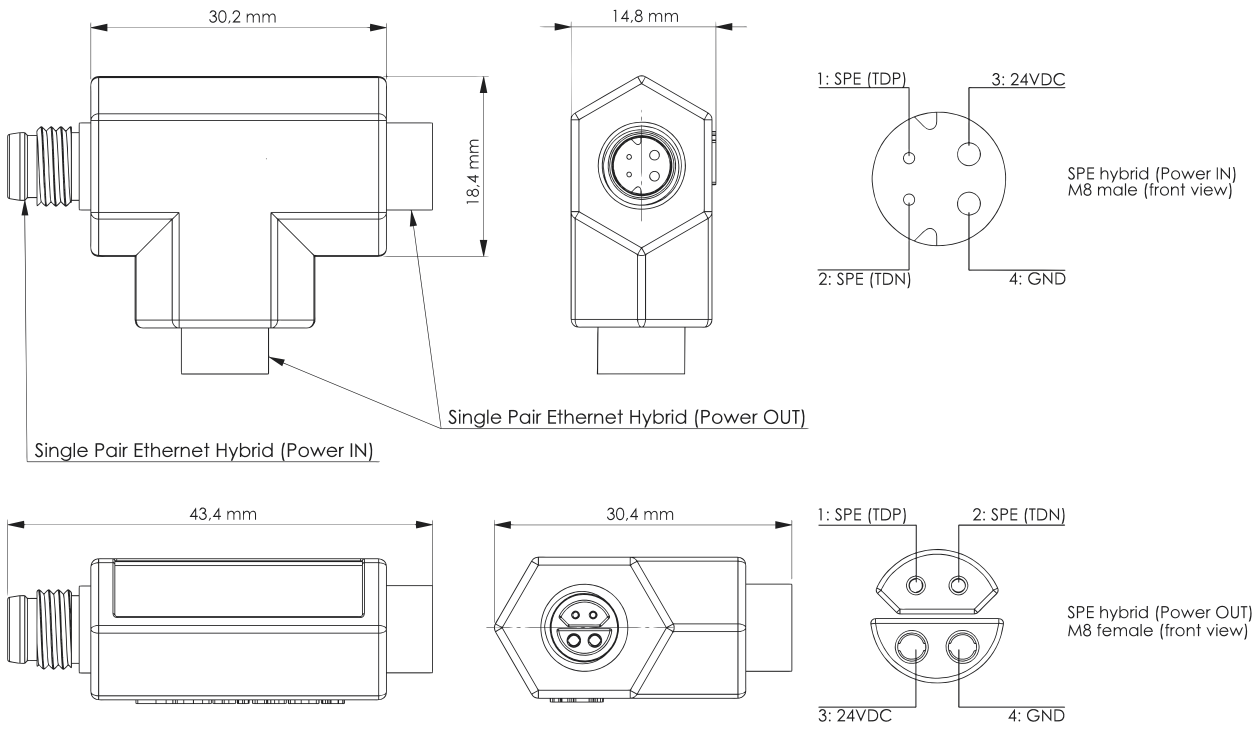

Figure 13: periSWITCH 3-port dimensional drawing & pinout

This switch significantly enhances the flexibility of cabling topologies, such as line and star configurations, among others. This adaptability makes it an ideal solution for a wide range of scenarios, encompassing everything from industrial production lines to commercial applications and smart buildings, to mention just a few examples.

Key Features:

- Features 1 SPE hybrid IN port for both power and data input.
- Includes 2 SPE hybrid OUT ports for distributing power and data.
- Supports series connections of devices.
- Enables efficient subdistribution at the field level.

periSWITCH 3-port interfaces:

Single Pair Ethernet Hybrid Interface (Power In)	
Connector type	M8 H-coded male connector according to IEC 63171-6:2021 (style 6J-M8CI without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC input
Input Voltage	24VDC (±10%)
Consumption	Average 0.5W (max. 2A troughput current)
Single Pair Ethernet Hybrid Interfaces (2 Power Out)	
Connector type	M8 A-coded female connector according to IEC 63171-6:2021 (style 6J-M8C without shielding)
Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)
Power	24VDC output

Table 21: periSWITCH 3-port interfaces

periSWITCH 3-port mechanical specifications

Mechanical Specification			
Housing	Dimensions	Length	43.4 mm
		Width	30.4 mm
		Height	14.8 mm
		Weight	20 grams
	Material		Thermoplastic material
	Protection		IP67

Table 22: periSWITCH 3-port mechanical specifications

periSWITCH 3-port electrical specifications:

Electrical Specifications			
Network Interfaces	Connector	Type	M8 H-coded male (Interface 1) M8 H-coded female (Interface 2/3)
		Standard	Similar to IEC 63171-6:2021 Style 6J-M8C
		Mating	Mates with periLINE and IEC 63171-6:2021 Style 6P-M8C
		Pin 1	TDP
		Pin 2	TDN
		Pin 3	24VDC Input
		Pin 4	GND
	Communication	Type	SPE
		Physical Layer	100BASE-T1
		Standard	IEEE 802.3bw
	Power	Voltage	24VDC
		Direction	Input (Interface 1) Output (Interface 2/3)
Alimentation	Voltage	Nominal	24VDC
		Range	±10%
	Consumption	Average	0.5 W

Electrical Specifications			
	Throughput current	Maximum	2A

Table 23: periSWITCH 3-port electrical specifications

Absolute maximum ratings:

Interface	Rating
Supply voltage	-40V – +40V
100BASE-T1	-40V – +40V

Table 24: periSWITCH 3-port absolute maximum ratings

Recommended Environment Conditions:

Operating conditions	-40°C – +70°C
Storage conditions	-45°C – +85°C

Table 25: Recommended environment conditions

Reverse Polarity Protection:

This smart component implements protection against polarity inversion for 24V DC input (Pin 3) and GDN (Pin 4), as well as TDP (Pin 1) and TDN (Pin 2) for the network interface.

Short Circuit limitation:

A protection mechanism against shorting the supply voltage to ground on the sensor or actuator interface might damage the device when current limitation is not applied. The current limitation of the power supply circuits is specified as max. 2A.

Note: It is important to be aware that the *periSWITCH 3-port* is not suitable for use in hard real-time or safety-critical applications.

Note: For optimal performance, the periSWITCH 3-port should be connected to other components using cables that are no longer than 15 meters in length.

2.4 periLINE Hybrid SPE Cable

Figure 14: periLINE

The periLINE hybrid Single Pair Ethernet (SPE) cable is an integral part of the Perinet Smart Components, designed to connect devices within the Seamless IoT Connectivity system. This cable is uniquely engineered to transmit both data and power, thereby streamlining and simplifying deployment processes.

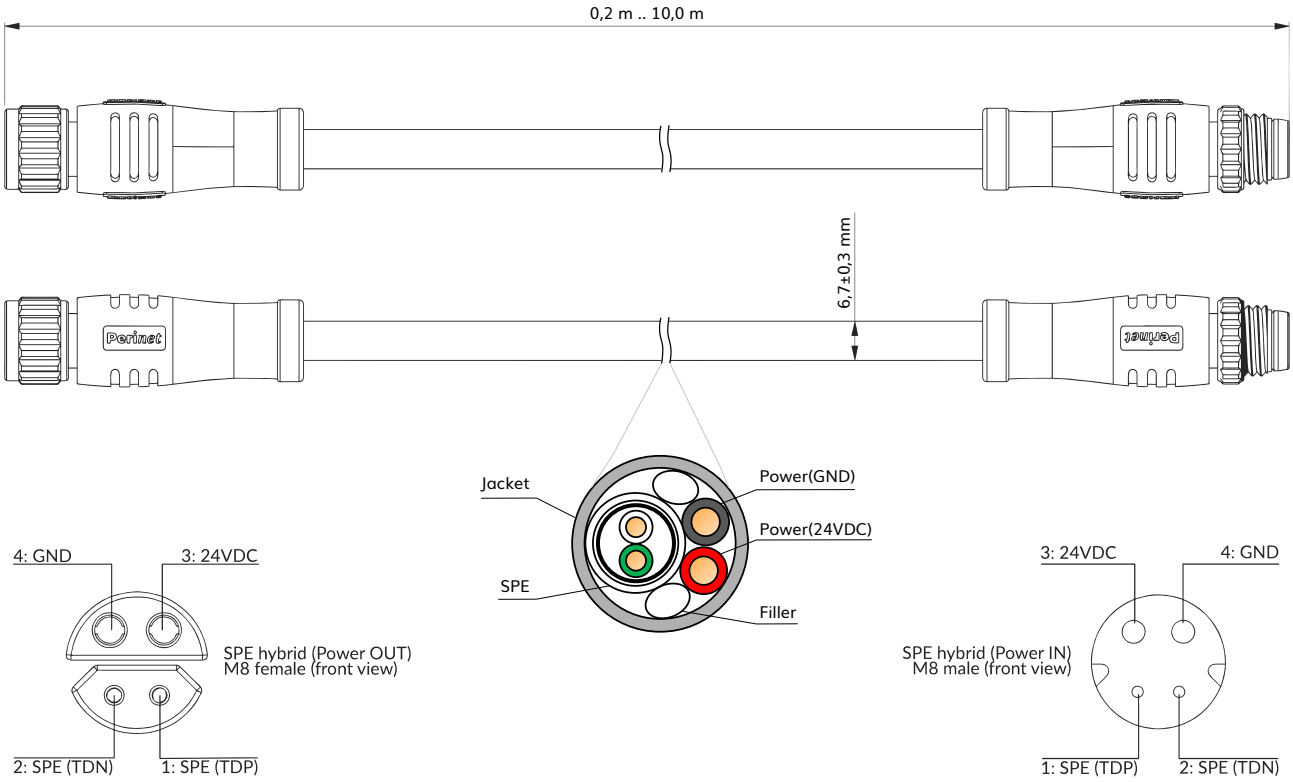

Figure 15: periLINE dimensional drawing & pinout

Connector Specification	
Type	M8 H-coded Hybrid SPE connector according to IEC 63171-6:2021 (style 6P-M8C/6P-M8CI without shielding)
Fixation	with free rotating screw nut
Connector Length	35 mm
Power Pin Diameter	1.0 mm
SPE Pin Diameter	0.5 mm
Contacts Material	Brass with 3µ" Gold Plated
Screw Nut Material	Brass with 120µ" Nickel Plated
Screw Nut Thread	M8 x 1mm (internal for female, external for male)
Insert Material	TPU + GF, black
Minimum Mating Cycles	200
Pin 1	SPE (TDP)
Pin 2	SPE (TDN)
Pin 3	Power (24VDC)
Pin 4	Power (GND)

Table 26: periLINE connector specification

Cable Specification	
Jacket Material	PUR, black
Jacket Outer Diameter	6.7 ±0.3 mm
Jacket Thickness	0.38 mm nominal
SPE Braiding	Tinned Copper, 85% coverage
SPE Al-mylar	100% coverage
SPE Insulation Material	PE (green for TDN, white for TDP)
SPE Insulation Outer Diameter	1.6 ±0.1 mm
SPE Insulation Thickness	0.45 mm nominal
SPE Conducter Material	Stranded Tinned Copper
SPE Conducter Thickness	26AWG
SPE Conductor Resistance	max. 150Ω/km at 20°C
Power Insulation Material	SR-PVC (red for 24VDC, black for GND)
Power Insulation Outer Diameter	1.45 ±0.1 mm
Power Insulation Thickness	0.23 mm
Power Conducter Material	Stranded Tinned Copper
Power Conducter Thickness	20AWG
Power Conductor Resistance	max. 36.7(Ω/km) at 20°C
Insulation Resistance	min. 100(MΩkm) at 20°C
Filler Material	Cotton
Minimum Bending Radius	25mm

Table 27: periLINE cable specification

General Specification	
Overall Length	variable from 0.2m to 15m (incl. connectors)
Degree of Protection	IP67 acc. to IEC 60529
Rated Voltage	300V
Rated Current	2A (for Power)
SPE Impedance	100±15(Ω, 1MHz)
SPE Communication	100BASE-T1 Single Pair Ethernet (IEEE 802.3bw)

Table 28: periLINE general specification

Assorted cable lengths available with the periLINE hybrid SPE cable offer versatility to meet a wide array of use cases and settings. This adaptability makes it suitable for diverse environments, ranging from industrial production lines to commercial applications and beyond.

Note: It is crucial to note that *periLINE* is not designed for use in hard real-time or safety-critical applications.

2.5 Perinet's Supported Network Topologies

2.5.1 Point to Point (PtP) - The Simplest Use Case

In a Point to Point (PtP) setup, the most basic and straightforward application scenario, only one media converter and one smart adapter are required. This configuration is sufficient to establish a direct connection of a single sensor or actuator to the existing infrastructure:

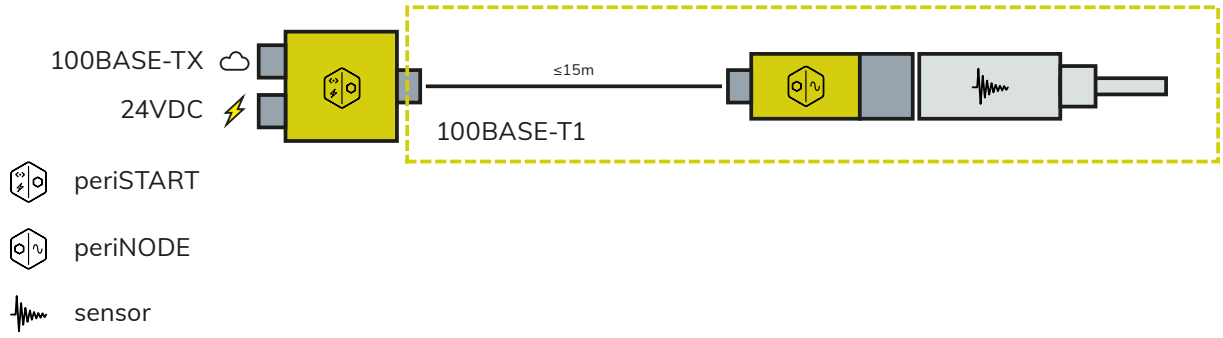

Figure 16: Point to Point Topology

2.5.2 Point to Multi-Point (PtMP)

In scenarios requiring a more complex setup, such as connecting multiple sensors and/or actuators to the existing infrastructure, Perinet provides various Point to Multi-Point (PtMP) topology options:

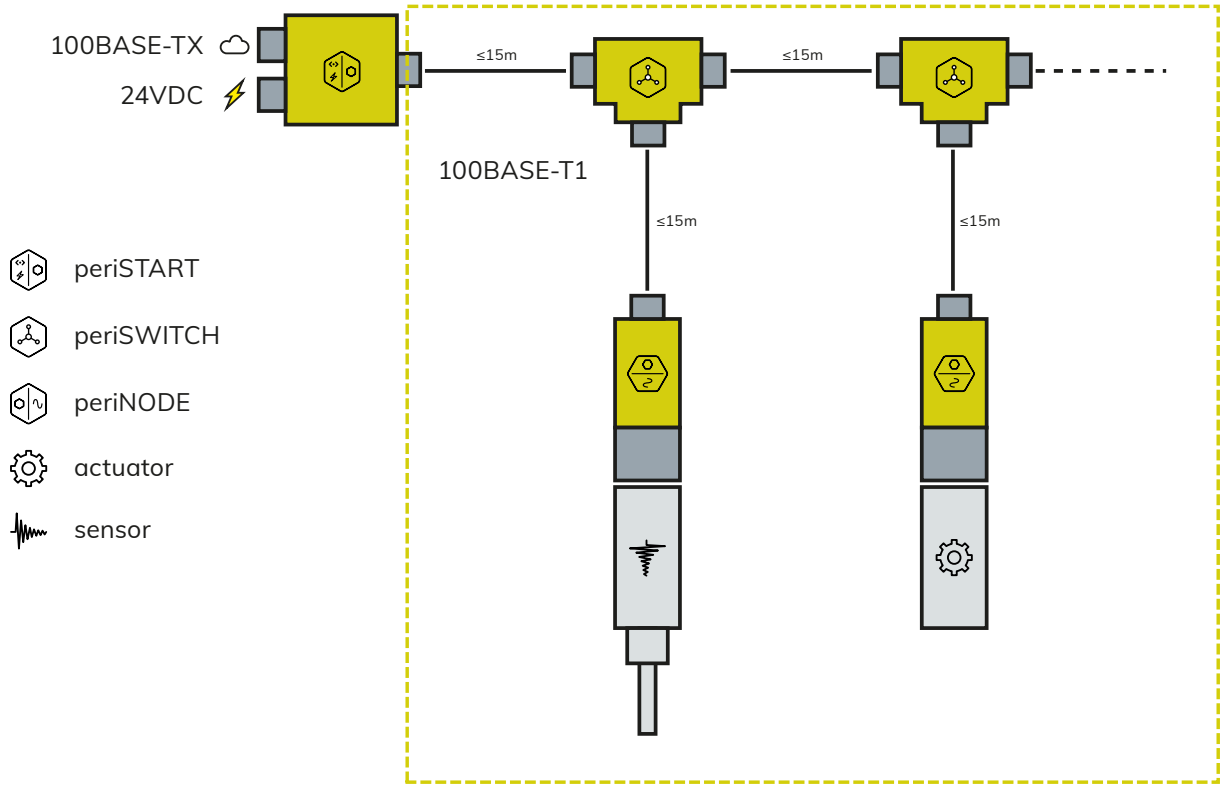

Figure 17: Line Topology

3 Smart Components Software Interface

Perinet Smart Components offer two main interfaces: a web-based user interface for humans and a RESTful API for machines. These interfaces are available through an integrated HTTP server, which requires TLS for secure access.

Note: Access to the integrated HTTP server is only possible with TLS enabled.

Each Smart Component has a unique hostname, formed by combining a host prefix with its unique serial number. Examples include `periNODE-serno`, `periSWITCH-serno`, and `periSTART-serno`, where `serno` is the serial number. This unique identifier is also present on the product's label (refer to Section 7 for details).

By default, Smart Components use the Zeroconf protocol multicast DNS (mDNS) and Link Local Multicast Name Resolution (LLMNR) to enable easy network discovery.

To access the web-based user interface, you can use mDNS-based name resolution. For instance, to access a periNODE device, you would use its mDNS name as follows:

```
https://periNODE-serno.local/
```

or via an LLMNR based name resolution:

```
https://periNODE-serno/
```

A RESTful API specification is available on each smart component via the URI:
`https://periNODE-serno.local/doc/api.proto`

3.1 Authenticity and Security Warnings

Users may encounter a security warning in environments where the **Perinet ECC Root CA** trust anchor is not installed. This certificate is essential for verifying *Perinet*'s Smart Components. It confirms that the component is genuinely made by an authorized manufacturer.

However, the **Perinet ECC Root CA** certificate is not automatically added to systems and requires manual installation. If this certificate is not present in the system, the authenticity of the Smart Component cannot be verified.

This situation occurs when a web client, like a standard web browser, attempts to establish a secure connection with the Smart Component. The web client first tries to authenticate the device. If the **Perinet ECC Root CA** certificate is missing, this authentication will fail, resulting in security warnings, as illustrated in Figures 18 (Edge browser) and 19 (Safari browser).

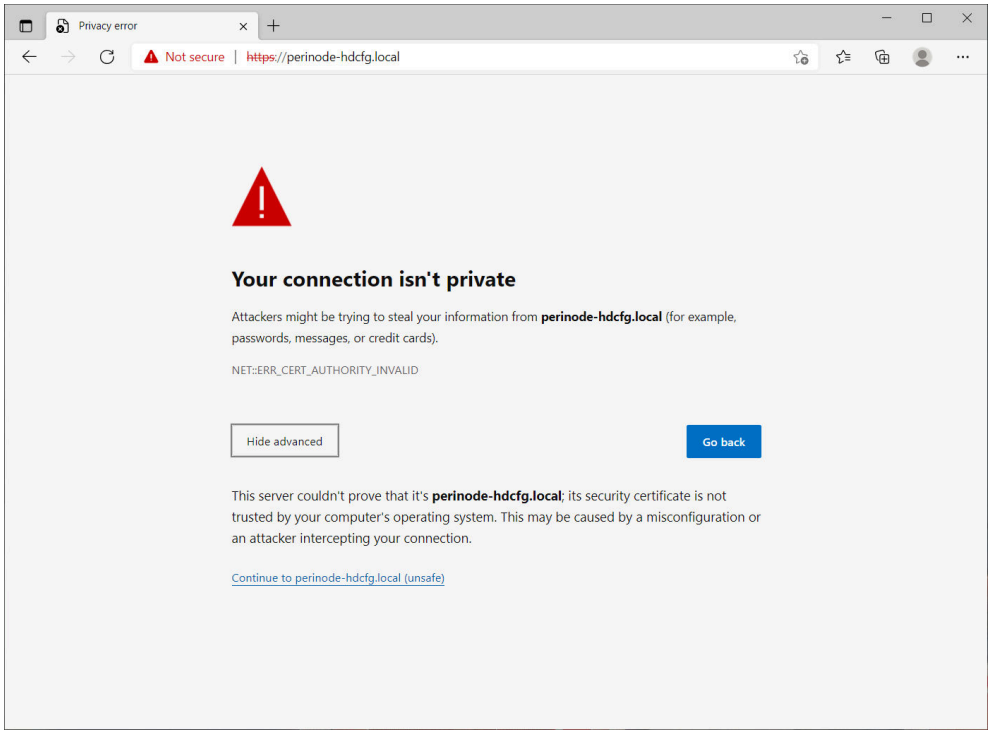

Figure 18: Edge browser security warning example

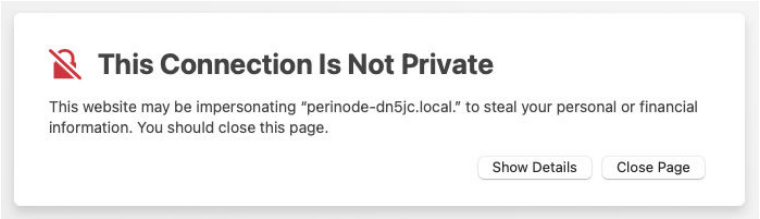

Figure 19: Safari browser security warning example

To address this issue, there are two options: either install the **Perinet ECC Root CA** trust anchor on the system, or choose to ignore the warning. However, bypassing the warning compromises system security.

Note: *Perinet* strongly advises installing the **Perinet ECC Root CA** in the system's list of trusted entities, instead of disregarding the security warning.

3.1.1 Add Perinet as a Trust Anchor

To install the **Perinet ECC Root CA** certificate, download it from https://docs.perinet.io. The installation process is straightforward: double-click the downloaded file and follow the on-screen instructions.

For comprehensive instructions on installing certificates across various operating systems and browsers, refer to the *Security Certificates Installation Guide* [4]. This guide assists in installing the necessary certificates for all *Perinet* Smart Components.

Once the **Perinet ECC Root CA** trust anchor is installed, you should be able to access the web user interface of any smart component without encountering security warnings. If issues persist, please reach out to our support team.

You can verify the installation by clicking on the *locker* icon in your web browser's URL bar. The subsequent figures demonstrate the expected certificate chain for reference.

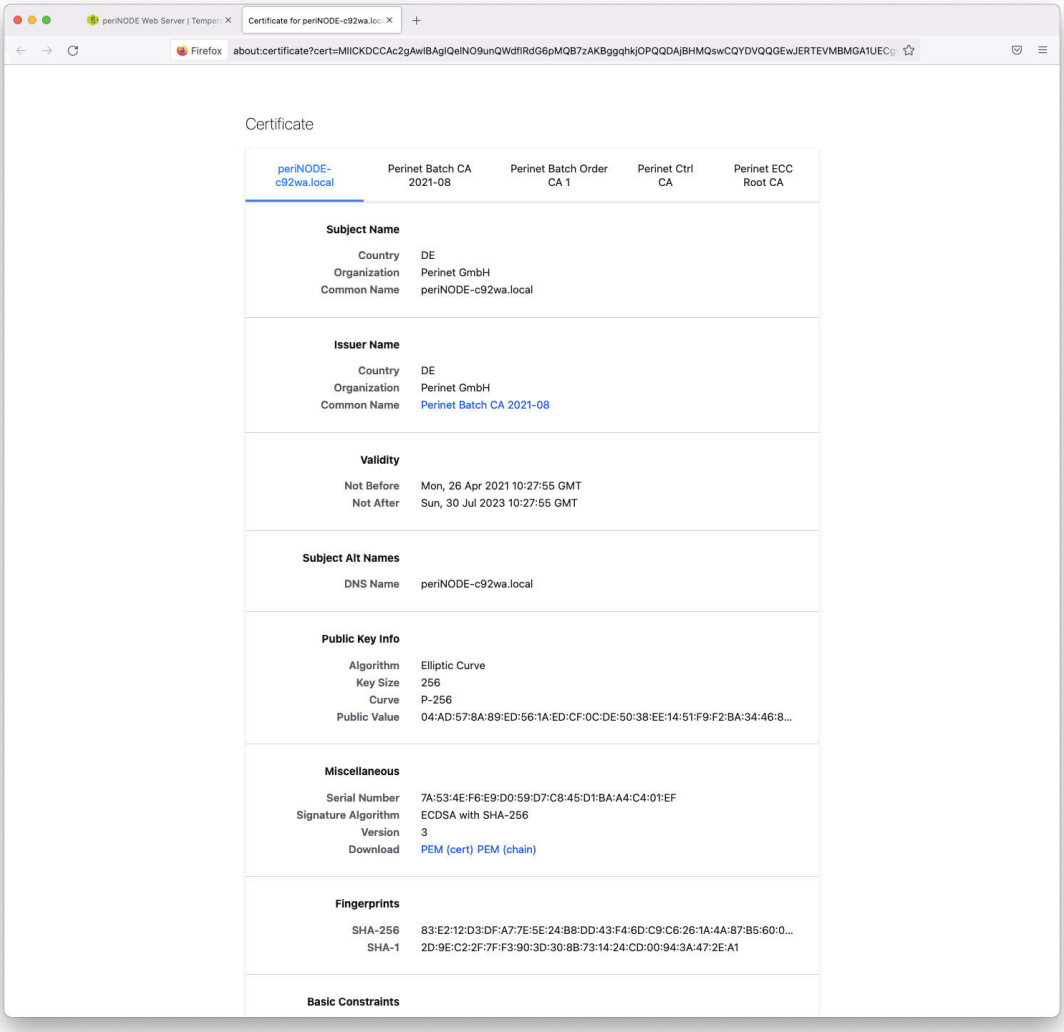

Figure 20: Firefox example security trusted chain

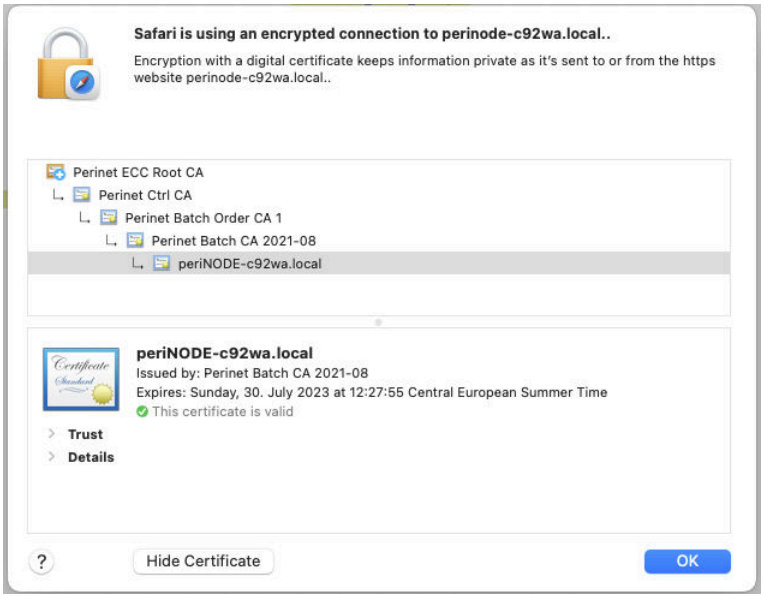

Figure 21: Safari example security trusted chain

Note: When using the RESTful API to access the Smart Component, the absence of the **Perinet ECC Root CA** trust anchor may impact command line tools. In this scenario, it's necessary to install the **Perinet ECC Root CA** trust anchor system-wide, rather than only in the web client (standard browser).

3.1.2 Proceed with Exception

As mentioned earlier, it's possible to bypass the security warnings if you choose not to add *Perinet* as a trusted entity. To do this, click on "Advanced" and confirm your intention to proceed. This action allows encrypted but unauthenticated communication.

When using the RESTful API to access a Smart Component, the absence of the **Perinet ECC Root CA** trust anchor can affect command line tools. However, there are ways to circumvent this. For instance, when using the **curl** tool, the `-k, -insecure` option allows proceeding without server authentication.

Note: *Perinet* strongly advises against ignoring security warnings. Without the certificate, the system is vulnerable to certain attacks, such as man-in-the-middle attacks.

3.2 Configuration

The firmware of the *periNODE* offers two configuration methods: a machine-friendly RESTful API and an HTML-based graphical user interface (GUI). The GUI is designed for interactive use, while the RESTful API is better suited for automated configurations.

All configuration settings are saved persistently and will be reinstated during the startup process.

Note: For *periSTART* and *periSWITCH* devices, users do not need to perform any software configuration.

3.2.1 periNODE Home

The *periNODE* Smart Component features a user-friendly interface for monitoring sensor measurements. The *Home screen* serves as the main page of the web user interface. An example for the *Pt100* variant is illustrated in Figure 22. To access the *Home screen*, use the following URL format: `https://periNODE-serno.local/`.

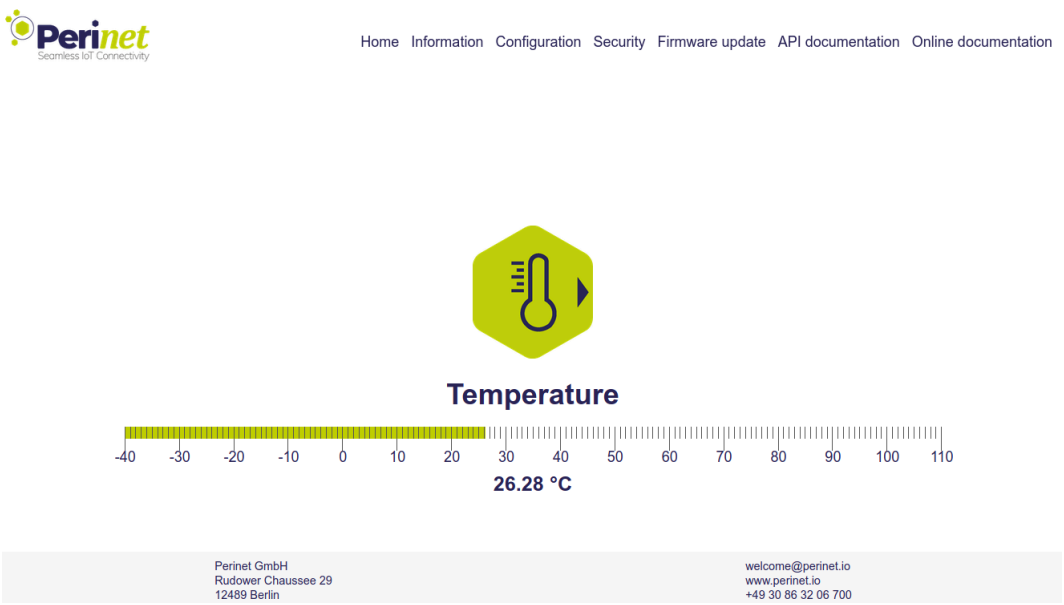

Figure 22: The Web Home Page of a periNODE Pt100 with attached temperature sensor

3.2.2 periNODE GUI Configuration

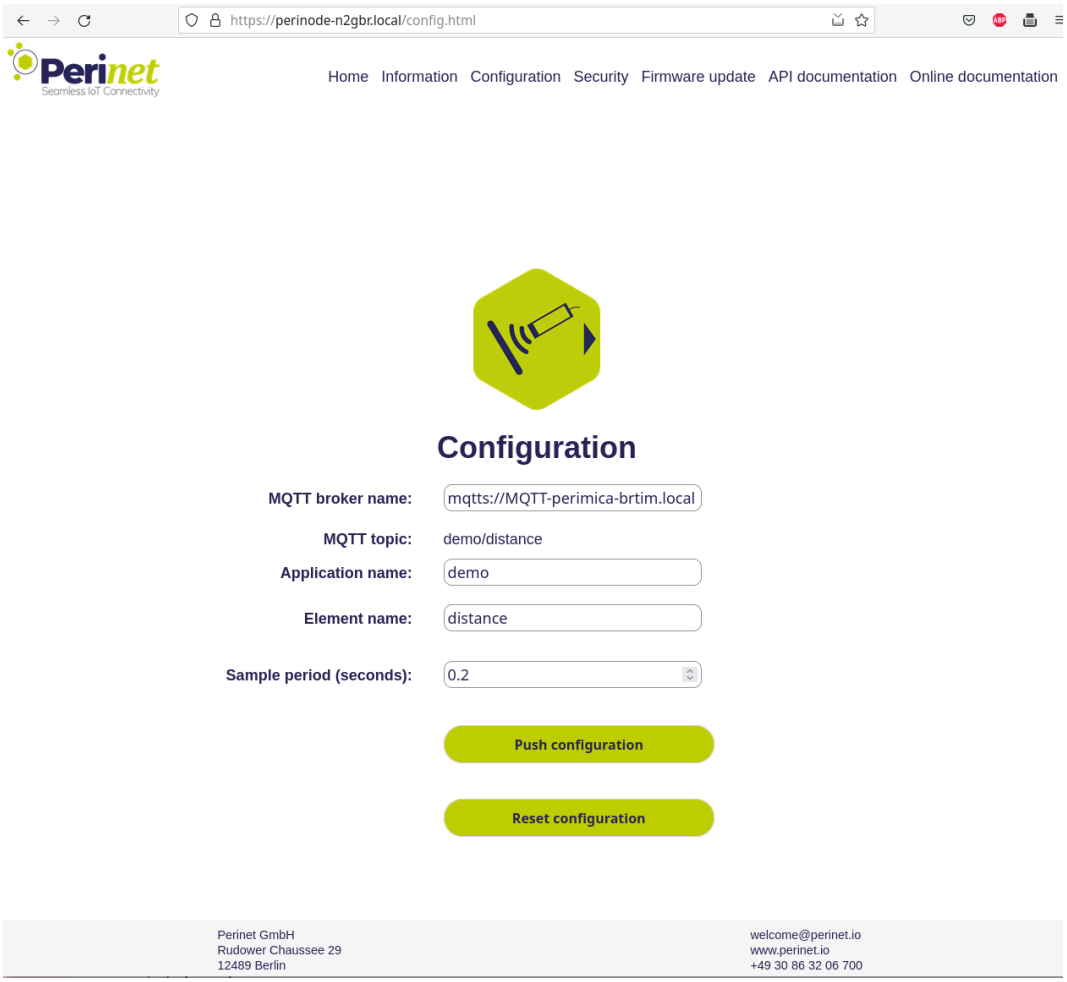

Figure 23: periNODE Configuration Web Page

The GUI (Graphical User Interface) of the *periNODE* enables configuration of several parameters:

- *MQTT broker name*: The (***mqtt_broker_name***) is used to connect to an MQTT broker **and shows a list of all locally discoverable MQTT brokers**.
- *Application name*: The (***application_name***) is used to group IoT devices and software components, that together represent an application. It also sets the prefix for MQTT topics.
- *Element name*: The (***element_name***) allows you to label things within applications understandably, getting rid of clumsy structural labeling like IP-addresses or hostnames.
- *Sample period (seconds)*: (***period_seconds***)
- *MQTT broker name*: The (***mqtt_broker_name***) parameter is essential for establishing a connection to an MQTT broker. **It additionally provides a list of all locally discoverable MQTT brokers**.

≥ FW 22

≥ FW 22

- *Application name*: The (***application_name***) is used to organize IoT devices and software components into a cohesive group, representing an application. This name also serves as a prefix for MQTT topics, ensuring a structured and identifiable messaging system.
- *Element name*: The (***element_name***) feature allows for the intuitive labeling of components within applications. This replaces the need for more complex identifiers like IP addresses or hostnames, simplifying system navigation and management.
- *Sample period (seconds)*: The (***period_seconds***) setting specifies the interval, in seconds, at which the system samples data. This parameter is crucial for controlling the data collection frequency, allowing for adjustments based on the application's performance and data requirements.

To apply a new configuration, click the ***Push configuration*** button. To reset to factory settings, use the ***Reset configuration*** button.

An example of these settings is shown in Figure 23. The ***application name*** is used to group IoT devices within the same application, organizing elements under a common umbrella. It also sets the prefix for MQTT topics, which *periNODE* smart adapters use to publish data. For instance, using the prefix ***demo/#*** in the MQTT protocol, you can subscribe to all sensor data related to the 'demo' application.

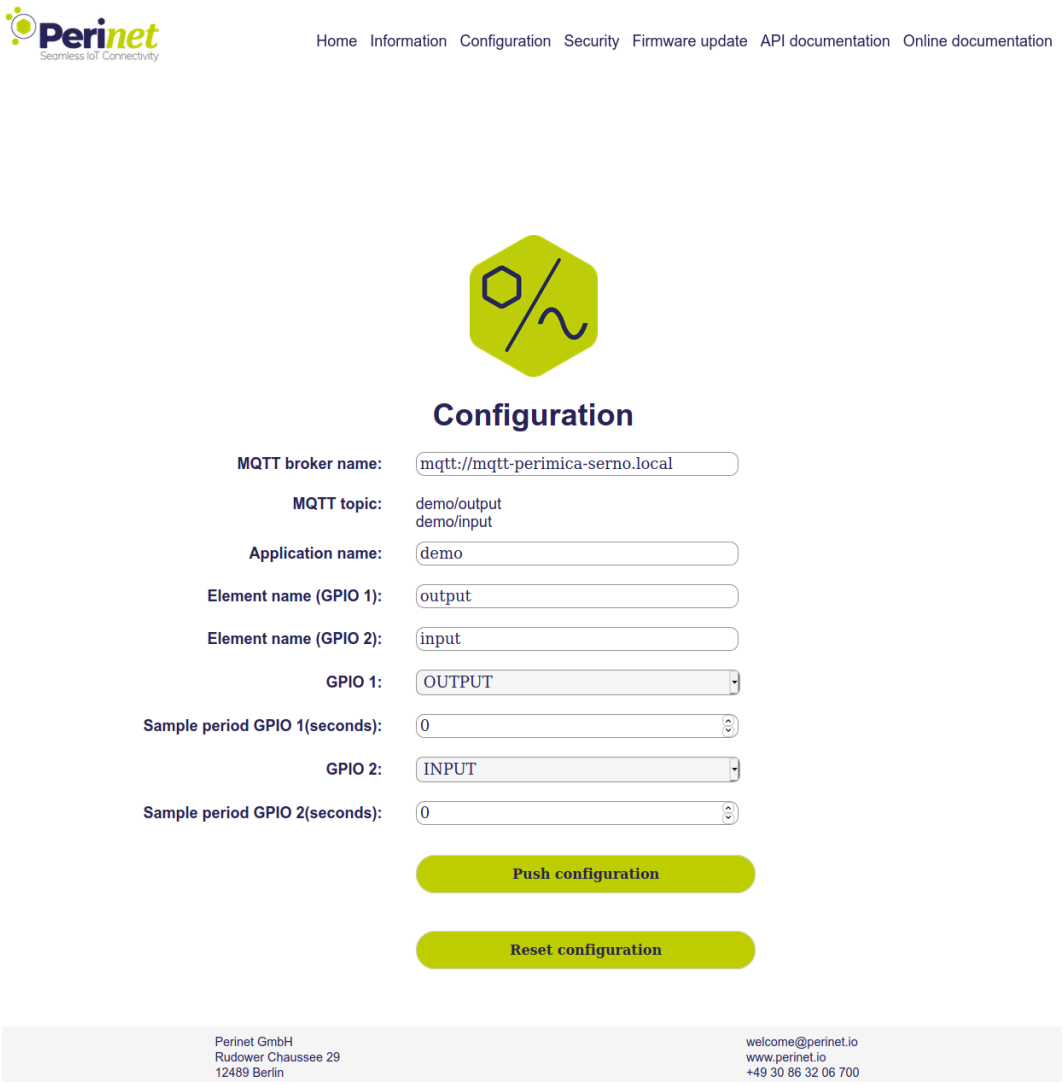

Figure 24: periNODE GPIO Configuration Web Page

The *periNODE GPIO* devices offer a unique configuration interface compared to other *periNODE* models. Besides the standard settings for the MQTT broker and application name, the *periNODE GPIO* firmware allows for detailed customization. This includes setting the element name for I/O ports 1 and 2 and designating each port's function as either ***INPUT*** or ***OUTPUT*** (see Figure 24 for more details).

Additionally, the firmware provides the ability to set a specific `sample period` for each port. This `sample period` is crucial as it defines the time interval between two consecutive data samples, directly influencing the sample rate. For detailed information on the range of `sample rate` values supported by this product, refer to Section 2.1.

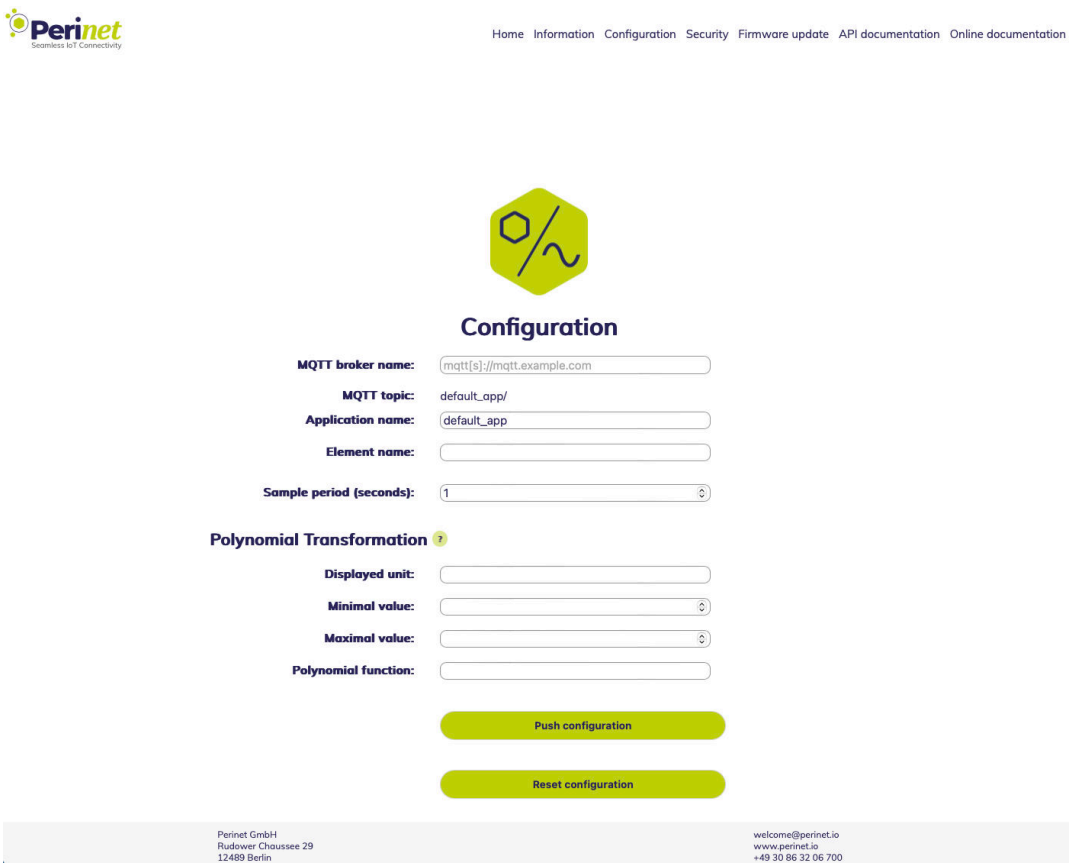

Figure 25: periNODE 0-10V Configuration Web Page

The *periNODE 0-10V* model includes a special feature in its configuration interface, known as the polynomial transformation. This function is crucial for using the device with various 0-10V sensors, guaranteeing precise and relevant data. The transformation involves four settings:

- *Displayed Unit*: The ***"displayed_unit"*** field specifies the output's measurement unit, like "mm" for millimeters.
- *Maximum Value*: The ***"max_value"*** field determines the highest output value. If exceeded, it shows "out of range".
- *Minimum Value*: The ***"min_value"*** field sets the lowest output value. Readings below this are displayed as "out of range".
- *Polynomial Function*: In the ***"polynomial_function"*** field, a mathematical formula is used to transform the 0-10V input into a specific measure. For example, a "10x" formula means the output will range linearly from 0 to 100.

3.2.3 periNODE RESTful API Configuration Information

Parameter name	*Description*
`mqtt_broker_name`	Specifies the address of the MQTT broker for connecting the firmware. Address formats include mDNS names (like mqtt-broker.local) or IPv6 link local addresses. For periNODE variants like GPIO, this setting enables the device to send sensor data or receive instructions. MQTT topics are formed as `<application_name>/<element_name>`.
`application_name`	A unique name that groups devices together. It forms the first part of the MQTT topic for publishing sensor data or subscribing to events. For more on IoT application setup, see [3].
`sink_interface.element_name`	Defines the MQTT topic suffix for controlling the *periNODE GPIO* actuator.
`source_interface.element_name`	Sets the MQTT topic suffix for publishing sensor data.
`sink-source_interface .period_seconds`	Time interval in seconds for sending MQTT messages.
`sink-source_interface .samples_per_period`	Number of sensor readings sent in the time frame set by ***period_seconds***.

Table 29: Configuration parameters

For details about sensor metrics through the RESTful API, see Section 3.2.4.

To access configuration values using the RESTful API, use this URL format:

```
https://periNODE-serno.local/config
```

This will provide the configuration in this JSON format:

```
{
    "application_name": "periNODE",
    "mqtt_broker_name": "mqtts://mqtt-perimica-serno.local",
    "sink_interface_config": {
        "element_name": "output",
        "period_seconds": 0.0,
        "samples_per_period": 1,
        "type": "DIGITAL_IO_SINK"
    },
    "source_interface_config": {
        "element_name": "input",
        "period_seconds": 0.0,
        "samples_per_period": 1,
        "type": "DIGITAL_IO_SOURCE"
    }
}
```


Listing 1: periNODE configuration object retrieved from the resource /config.

The excerpt below demonstrates how to get configuration data from a periNODE using the command line tool `curl`:

```
curl -X GET https://periNODE-serno.local/config
```

To modify configuration settings, a HTTP PATCH request to `/config` is needed. This requires sending a valid JSON object. It's not necessary to send the entire configuration object. Only the included attributes will be updated in the periNODE's configuration. The example below shows how to change the `mqtt_broker_name` in a periNODE using ***curl***:

```
curl --data \
 '{"mqtt_brocker_name":"mqtts://mqtt-perimica-serno.local"}'\
 -X PATCH https://periNODE-serno.local/config
```

Note: Remember to replace the host network interface and serial number in the URL with the actual serial number of your periNODE smart adapter.

For Linux systems, especially those using Debian distributions, install ***curl*** with this command:

```
apt install curl
```

For other operating systems, you can download ***curl*** from https://curl.se/download.html.

3.2.4 periNODE RESTful API Sample Information

To get sensor input details or modify actuator output settings, sample data can be accessed through the RESTful API. This data is provided in a JSON format. You can obtain this information from the `/sample` resource of a periNODE or it might be available on the designated MQTT broker.

Key	*Value/data type*	*Description*
incarnation	number/integer	Number that increases with each reset/reboot of the controller and after the overflow of the *sequence_number*
sequence_number	number/integer	Message counter if sample information has changed or was changed by the user.
data	array	Data segment of the payload, repeating group representing the actual sensor or actuator data.
unit	string	Unit of the <value>
<value> e.g. distance	bool, integer, float…	Sensor or actuator value

Table 30: Sample attributes

For RESTful API information related to configuration settings, see Section 3.2.3.

To retrieve the JSON formatted sample data, use the following example:

```
curl -X GET https://periNODE-serno.local/sample
```

```
{
    "data": [
        {
            "temp": 22.89,
            "unit": "°C"
        },
        {}
    ],
    "incarnation": 8,
    "sequence_number": 35727
}
```


Listing 2: JSON encoded sample object of a periNODE Pt100

To modify the state of an actuator, send a JSON object with an HTTP PATCH request to the same resource.

3.2.5 periNODE RESTful API Sample Information for periNODE-GPIO

Unlike other Smart Components, the *GPIO* variant of periNODE features two interfaces in one unit. These can be set up as either sensors (GPIO input) or actuators (GPIO output). This dual functionality leads to a unique approach in handling *sample* data.

The *GPIO* version supports three REST endpoints:

- `/sample/gpio1`: Accesses the first GPIO.
- `/sample/gpio2`: Accesses the second GPIO.
- `/sample`: Provides data for both GPIOs.

An example on how to retrieve the JSON encoded sample object for both GPIOs is shown in Listing 3:

```
curl -X GET https://periNODE-serno.local/sample
```

```
{
    "data": {
        "GPIO1": false,
        "GPIO2": false
    },
    "incarnation": 114,
    "sequence_number": 524
}
```

Listing 3: JSON encoded sample object of a periNODE GPIO for /sample endpoint

An example on how to retrieve the JSON encoded sample object for one dedicated GPIO is shown in Listing 4:

```
curl -X GET https://periNODE-serno.local/sample/gpio1
```

```
{
    "data": false
    "incarnation": 114,
    "sequence_number": 536
}
```

Listing 4: JSON encoded sample object of a periNODE GPIO for /sample/gpio1 endpoint

To alter an actuator's state, you need to send a JSON object using an HTTP PATCH request to the same resource. The example below demonstrates how to activate the output of GPIO 2 through the RESTful API:

```
curl -6 -g -k --interface eth0 \
 --data '{"data":true}'\
 -X PATCH https://periNODE-serno.local/sample/gpio2
```

Sending a partial object with the setting `true` for GPIO 2 turns it on, equivalent to applying 24V. Setting it to `false` turns it off, equivalent to 0V.

Note: The HTTP PATCH method cannot be used with the `/sample` resource.

Note: The JSON object sent to the periNODE is different from the JSON object received from it.

3.3 Security Configuration

Security settings for a periNODE can be modified using the RESTful API at `/security` or through a web interface accessible at *https://<hostname>.local/security.html*.

There are three certificates for configuring a periNODE's security:

- ***Host Certificate***: A unique certificate that validates the periNODE's identity to external clients like web browsers. It includes the hostname and must be signed by a trusted root CA, ensuring the periNODE's authenticity.
- ***Root Certificate***: Represents the trusted authorities in the periNODE system. It is essential for verifying external clients (e.g. web browsers) during connections, especially when mutual TLS (mTLS) is enabled.
- ***Client Certificate***: Utilized by the periNODE when connecting as a client to external servers, like MQTT brokers. This certificate authenticates the periNODE **and defines its access level in these interactions, under the Role Based Access Control (RBAC) system**.

≥ FW 22

For more detailed information on security concepts used for Perinet products, please refer to https://docs.perinet.io.

3.3.1 Mutual TLS and Authorization

A periNODE can be set up to use mutual TLS (mTLS) and Role Based Access Control (RBAC). With mTLS enabled, any remote client must authenticate itself using a client certificate to establish a connection. This certificate is checked against the periNODE's *Root Certificate*.

Note: Remote clients can be authenticated by the periNODE only if the *Root Certificate* is stored in the periNODE and the *Client Certificate* is signed by this same *Root Certificate*.

A user role must be included in the client certificate for the *periNODE*. This role is crucial for the Role Based Access Control (RBAC) system. The *periNODE* supports three roles: ***admin***, ***super***, and ***user***:

- ***admin***: This user has complete read/write access to the *periNODE*, with no limitations.
- ***super***: This user can access and modify all resources, except for */security* and */update*.
- ***user***: This user can only view resources and cannot modify any.

Note: The ***user*** role is not permitted to modify actuator states via the RESTful API.

Mutual TLS (mTLS) is not active by default. It will also be deactivated following a security or factory reset.

Note: By default, role-based access control for both the RESTful API and web user interface is *deactivated*.

For more detailed information on the client certificate, please refer to https://docs.perinet.io.

3.3.2 Web User Interface

The web user interface includes sections for inputting all three types of certificates: the *Host Certificate*, *Root Certificate*, and *Client Certificate*. For illustration, Figure 26 displays the input section specifically for the *Host Certificate*.

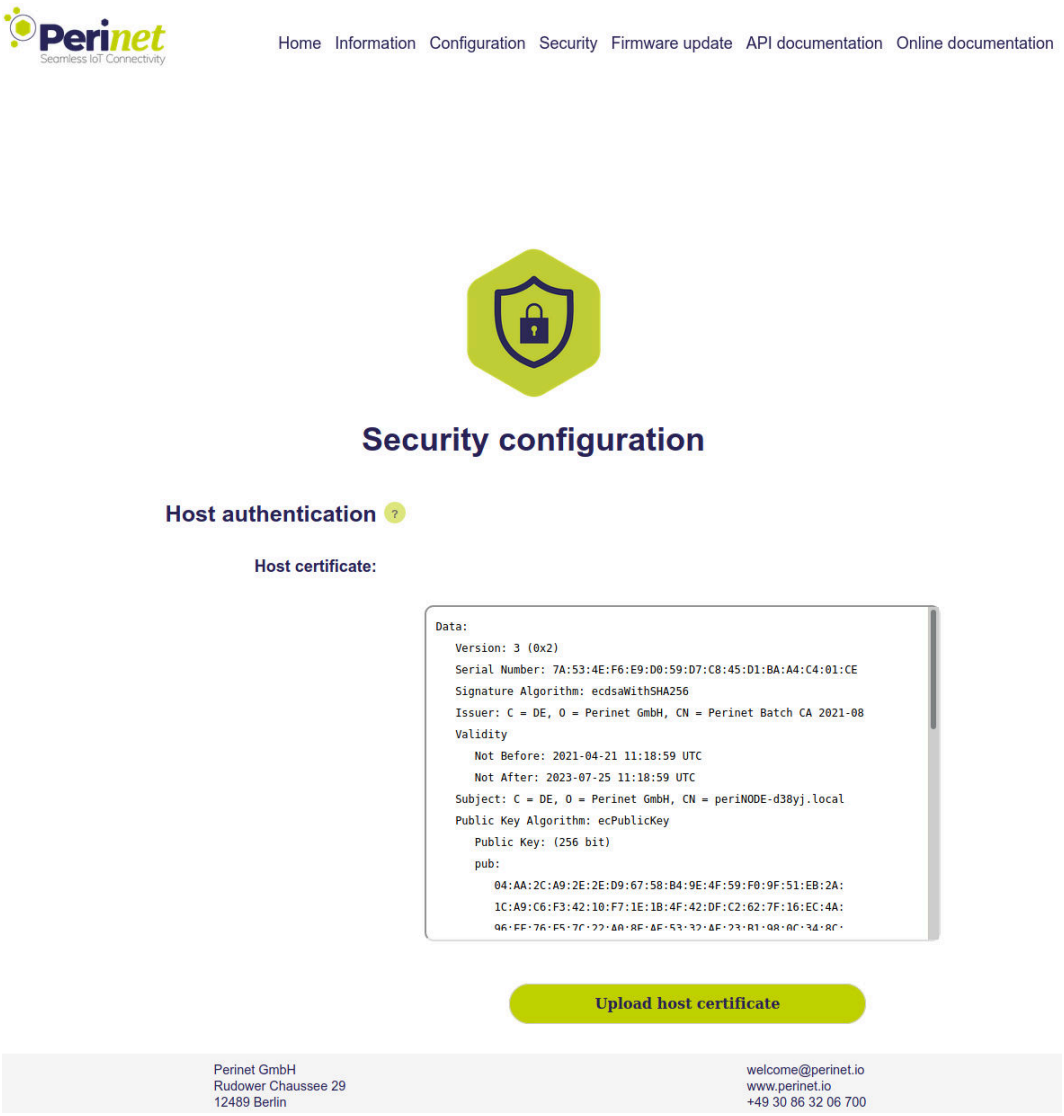

Figure 26: Security web user interface

The certificate displayed in the text area is the current one stored in the *periNODE*. This device accepts X.509 certificates encoded in PEM format (Base64 ASCII). Commonly, files with the extension `.pem` use this encoding, but it's also seen in `.crt`, `.cer`, and `.key` files.

For the *Host Certificate* and the *Client Certificate*, they should be uploaded along with their respective private keys concatenated at the end. Conversely, the *Root Certificate* should be uploaded without its private key.

For detailed guidance on generating certificates, visit https://docs.perinet.io.

Enforce mTLS Access

Activating the mTLS feature mandates that all remote clients authenticate themselves to the *periNODE* using a valid *Client Certificate*. This *Client Certificate* is verified against the *Root Certificate* already stored in the system.

Note: Before activating 'Enforce mTLS Access', ensure that a valid *Root Certificate* is already in place.

3.3.3 RESTful API Certificate Deployment

The RESTful API facilitates the management of all three types of certificates through dedicated resources. These resources are modifiable exclusively by the *admin* role and must be updated using an HTTP `PATCH` request. The specific endpoints for each certificate are as follows:

- *Host Certificate*: Accessed via `/security/host-cert`.
- *Root Certificate*: Accessed via `/security/root-cert`.
- *Client Certificate*: Accessed via `/security/client-cert`.

The subsequent section provides an example of updating the *Root Certificate* using the **curl** command-line tool.

```
curl --data-binary @<certificate-file> \
 -H "Content-Type: application/octet-stream" \
 -X PATCH https://periNODE-serno.local/security/root-cert
# reboot device
curl -X PATCH https://periNODE-serno.local/reboot
```

'Enforce mTLS Access' can be activated through the RESTful API, as demonstrated in the upcoming excerpt:

```
curl --data='{"enable_user_role":true}' \
  -X PATCH https://periNODE-serno.local/security
# reboot device
curl -X PATCH https://periNODE-serno.local/reboot
```

3.3.4 Reset Security Configuration

The **Reset security** button enables users to revert the security settings to their original factory defaults. This action will replace any user-configured Host and *Root Certificates* with the default ***Perinet ECC Root CA*** certificates. Additionally, any existing *Client Certificate* will be removed.

The procedure to reset the security configuration via the **curl** command-line tool is shown in the following code snippet:

```
curl -X PATCH https://periNODE-serno.local/security/reset
```

If the *periNODE* becomes inaccessible, such as in cases where the *Client Certificate* is lost, please consult the Factory Reset procedure detailed in Section Section 3.7.

3.4 MQTT

The application firmware of the periNODE supports the MQTT protocol to publish sensor values and/or subscribe to an information stream, which can be used to control actuators. The periNODE is compatible to MQTT protocol version 3.1.1.

The periNODE firmware is able to detect MQTT brokers through an mDNS based service discovery protocol (DNS-SD). Brokers are discovered by browsing for the ***service_type*** `_mqtt._tcp` (or `_secure-mqtt._tcp` for MQTT brokers using TLS) and the hostname of the broker e.g. “mqtt-perimica-serno”. Discovered brokers can be viewed in the MQTT broker text field drop-down list of the periNODE configuration Web UI. MQTT brokers which do not have the ability to advertise themselves via the mDNS based service discovery protocol can be configured manually via the web user interface or RESTful API.

As the periNODE firmware only supports IPv6, the local network and the broker must also be capable of communicating via IPv6.
The MQTT topic under which the periNODE smart adapter publishes messages or subscribes to is based on the ***application_name*** and ***element_name*** provided by the configuration parameters:

```
topic :  "<application_name>/<element_name>"
```

The MQTT message payload is in a JSON encoded format. After configuring the periNODE with the desired broker, it is possible to subscribe to periNODE sensor data using any MQTT client, as the client that is provided by the ***mosquitto*** library.

MQTT Client - Command Line Examples

If the configured ***application_name*** of the device is ***measure***, the ***element_name*** is ***distance*** and the configured MQTT broker is ***mqtt.local***, one can subscribe to the periNODE messages (e.g. using the ***mosquitto*** library):

```
mosquitto_sub -h mqtt.local -t measure/distance"
```

which prints sample JSON messages similar to the following:

```
{"incarnation":2, "sequence_number":178, "interface_type":"DISTANCE_SOURCE",
{"data":[{"unit":"mm","distance":311}]}
```

Or:

```
mosquitto_sub -h mqtt.local -t "measure/contactor"
```

which prints sample JSON messages similar to the following:

```
{"incarnation":120,"sequence_number":4,"data":true}
```

3.5 Secure MQTT

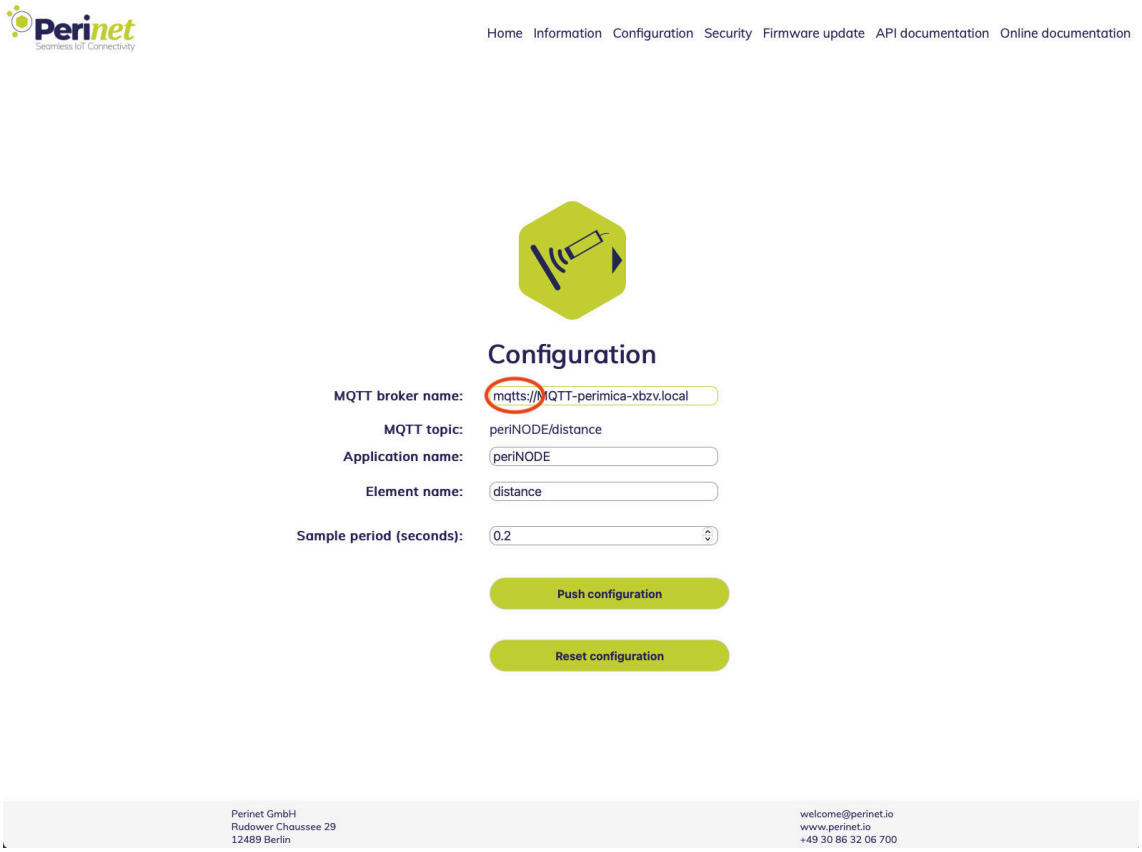

Figure 27: Configure secure MQTT

The communication between periNODE and MQTT broker can also be protected through TLSv1.2. The periNODE accesses the MQTT broker, verifying the broker certificate against the root certificate configured in the security page. Therefore, the periNODE MQTT client uses the same root certificate as the periNODE HTTPS server to authenticate the remote entity.

Mutual TLS (mTLS), where the remote MQTT broker authenticates the periNODE before continuing with the connection request, is supported as well. The periNODE uses the configured *client certificate* (see Section 3.3) in order to authenticate itself towards the MQTT broker.

Secure MQTT on the periNODE smart adapter can be configured implicitly in the configuration page of the periNODE web GUI (Figure 27), by inserting the ***mqtts://*** prefix in front of the ***MQTT broker name***. Leaving out the prefix at all or using ***mqtt://*** as prefix will disable TLS entirely for the MQTT client connection, which is not recommended by Perinet.

Mosquitto Examples using mTLS:

Subscribing to periNODE sensor data:

```
mosquitto_sub -h mqtt-secure.local -t test/topic -p 8883
--cafile root-ca.crt --cert client.pem --key client.key
```

Publishing messages:

```
mosquitto_pub -h mqtt-secure.local -t test/topic -p 8883 -m testmessage
--cafile root-ca.crt --cert client.pem --key client.key
```

3.6 Firmware Update

Perinet may provide security and/or feature updates for the smart components over time. Those updates will be delivered for each periNODE variant as an individual artifact. A firmware update shall be performed with the proper firmware update image artifact (e.g. matching variant).

Note: Inter-variant firmware updates are not supported by Perinet and may result in damaging the product.

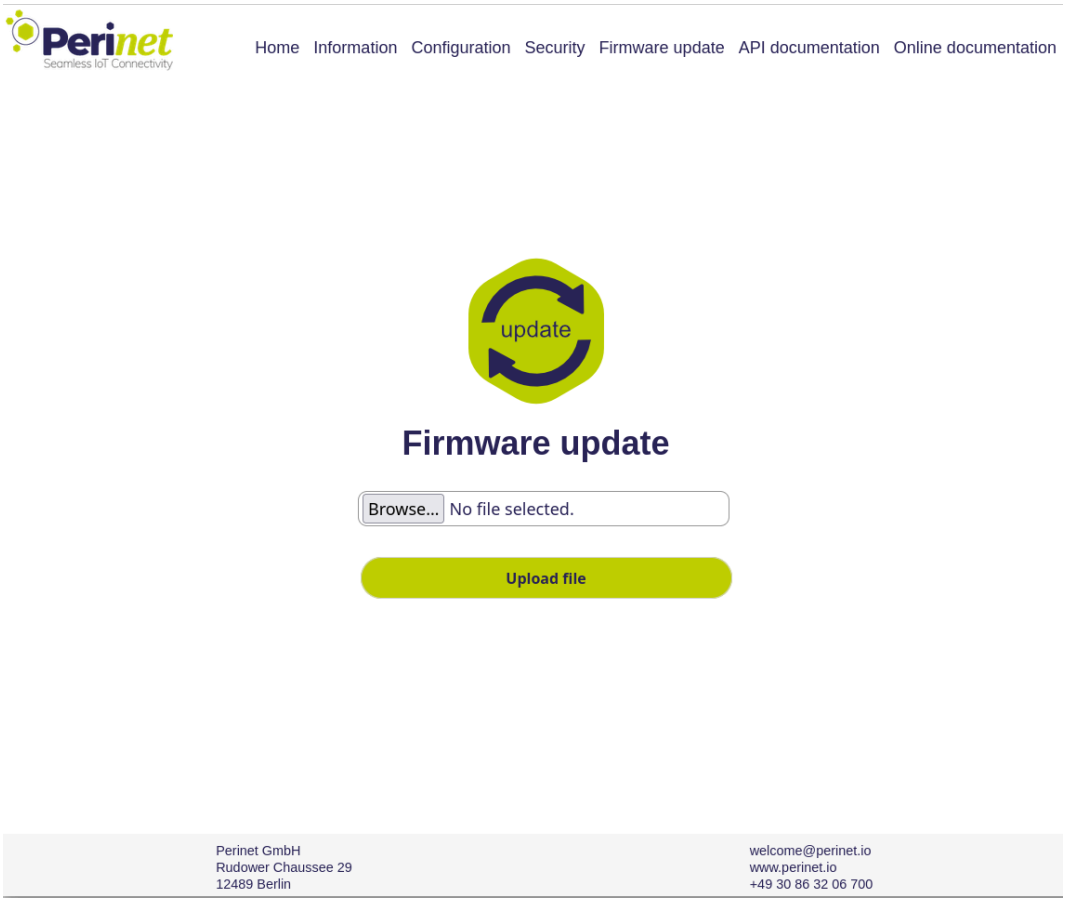

Figure 28: Firmware update

Note: A firmware update performed with the **Firefox** browser has been observed to corrupt the firmware image during the upload and result in a missing web user interface. A periNODE would still be functional and accessible via the RESTful API and accept firmware images via the RESTful API.

In order to keep the firmware of the device up to date, follow these steps:

1. Download the newest version of the firmware matching you product from https://downloads.perinet.io

2. Access the web page of the component through your browser:
   `https://hostname.local/`
3. Click on the **Firmware update** link located in the header of the page.
4. On the update page, click on **Browse...** and select the new firmware image to be uploaded.
5. Finally click on the **Upload file** button to install the new firmware.

Note: After a successful firmware update, the reboot of the component to the newly installed firmware will only be automatically triggered when using the web user interface.

3.6.1 RESTful API

The RESTful API accepts an update image via a *PUT* request on the resource */update*. The following command line shows how to perform an update with the **curl** tool:

```
curl --data-binary @<filename> \
    -H "Content-Type: application/octet-stream" \
    -X PUT https://periNODE-serno.local/update
# reboot into new firmware
curl -X PATCH https://periNODE-serno.local/reboot
```

3.7 Factory Reset

A factory reset of a smart component can be performed via the RESTful API remotely or physically by using a dedicated reset cable. The former solution might not be possible in some cases, e.g. where the client certificate has been lost and the access to the device is not possible anymore.

Factory Reset via Reset Cable

Connect the smart component via the reset cable (see Figure 29) for at least 20 seconds to a powered network, e.g. a powered periSTART. Please ensure that the power supply of the network is active before connecting the product. When resetting a periSWITCH or a periSTART, ensure that only one network port of the product is connected.

Figure 29: Perinet Reset Cable for Factory Reset

Factory Reset via RESTful API

When performing the factory reset via the RESTful API, a config as well as a security reset needs to be performed:

```
curl -X PATCH https://periNODE-serno.local/config/reset
curl -X PATCH https://periNODE-serno.local/security/reset
```

4 Supported Web Browsers

Smart Components feature a basic, integrated web server to display information and facilitate device configuration. For optimal experience, it is recommended to access these web pages using the following browsers:

- Windows Edge
- Safari

Important Notes:

- Currently, our firmware does not support Firefox due to an existing bug introduced from a Firefox firmware update.
- Google Chrome is incompatible with Perinet Smart Components when using Link Local Address IPv6 without an internet connection.

5 Qualifications and Approvals

5.1 CE Compliance

Figure 30: CE sign

Perinet products are CE compliant. They meet EU safety, health and environmental protection requirements.

5.1.1 Electromagnetic Compatibility (EMC)

Perinet products comply to the harmonized Immunity Standard for industrial environments (EN 61000-6-2:2005, EN 61000-6-2:2005/AC:2005), as well as to the harmonized Emission Standard for industrial environments (EN 61000-6-4:2007, EN 61000-6-4:2007/A1:2011).

Perinet Smart Components are designed to operate reliably within their electromagnetic environment as per the EMC Directive 2014/30/EU. This means they neither disrupt nor are affected by other devices in this environment.

These products adhere to the harmonized standards for immunity and emission in industrial environment. Specifically, they comply with the Immunity Standard (EN 61000-6-2:2005 and EN 61000-6-2:2005/AC:2005) and the Emission Standard (EN 61000-6-4:2007 and EN 61000-6-4:2007/A1:2011).

5.1.2 Environmental Compatibility

5.1.3 Environmental Compatibility

Perinet products meet the environmental standards set by IEC 61131-2:2007 for programmable controllers. These products are designed to operate effectively in temperatures ranging from -40°C to 70°C.

Additionally, Perinet Smart Components have been tested and passed the environmental protection tests for enclosures, as specified in EN 60529:1991 + A1:2000 + A2:2013.

- IP-Code x5: jetting water
- IP-Code x7: temporary immersion
- IP-Code 6x: dust-tight (with underpressure)

With regards to

EN 62368-1:2014/AC:2015 Audio/video, information and communication technology equipment - Part 1: Safety requirements

Perinet Smart Components are not covered by the previously mentioned standard.
They are intended exclusively for use in local installations that have common grounding. It's important to note that no lines from these devices should be directly connected to circuits that exit the building where the electronic equipment is installed.

5.1.4 RoHS and REACH

Figure 31: RoHS sign

Perinet products adhere to RoHS and REACH regulations, ensuring they are free from hazardous substances. Specifically, these products do not contain materials like lead, cadmium, mercury, hexavalent chromium, polybrominated biphenyls (PBB), polybrominated diphenyl ethers (PBDE), and certain phthalates (DEHP, BBP, DBP, DIBP).

Additionally, all Perinet products comply with the substance-related mandates of REACH. Our suppliers actively ensure that these products are devoid of substances of very high concern (SVHC) as identified by REACH.

Perinet GmbH assures that, under normal use and conditions, the goods supplied are not chemical products in nature.

5.1.5 Waste from Electrical and Electronic Equipment (WEEE)

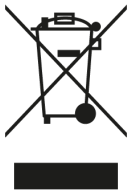

Figure 32: WEEE sign

Perinet products comply with the WEEE directive, adhering to EU regulations for the disposal and recycling of waste electrical and electronic equipment. This compliance is part of our commitment to promoting sustainable production and consumption practices.

6 Installation

The installation design for Perinet Smart Components involves securely mounting the sensors and providing strain relief for the cables. These Smart Components are not equipped with mounting features like holes or screw threads. Due to their compact size, they can be effectively positioned and supported by the sensors and cables themselves.

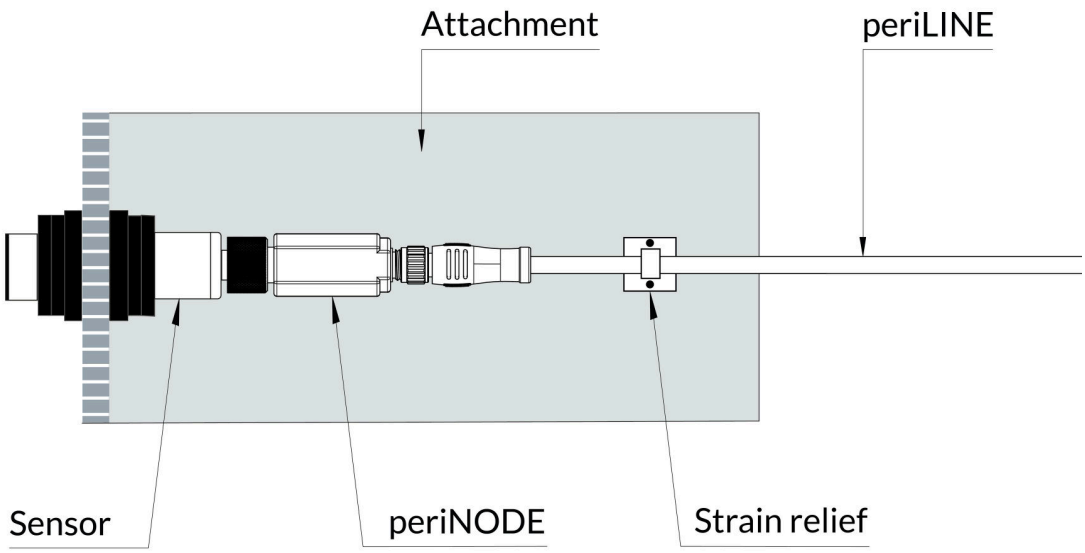

Figure 33: periNODE installation

The periNODE 0-10V and periNODE Pt100 can be securely attached by plugging them into the mounted sensor using the M12 circular connector. To avoid damage from transversal forces, it's important to fix the periLINE using a strain relief to a stable structure.

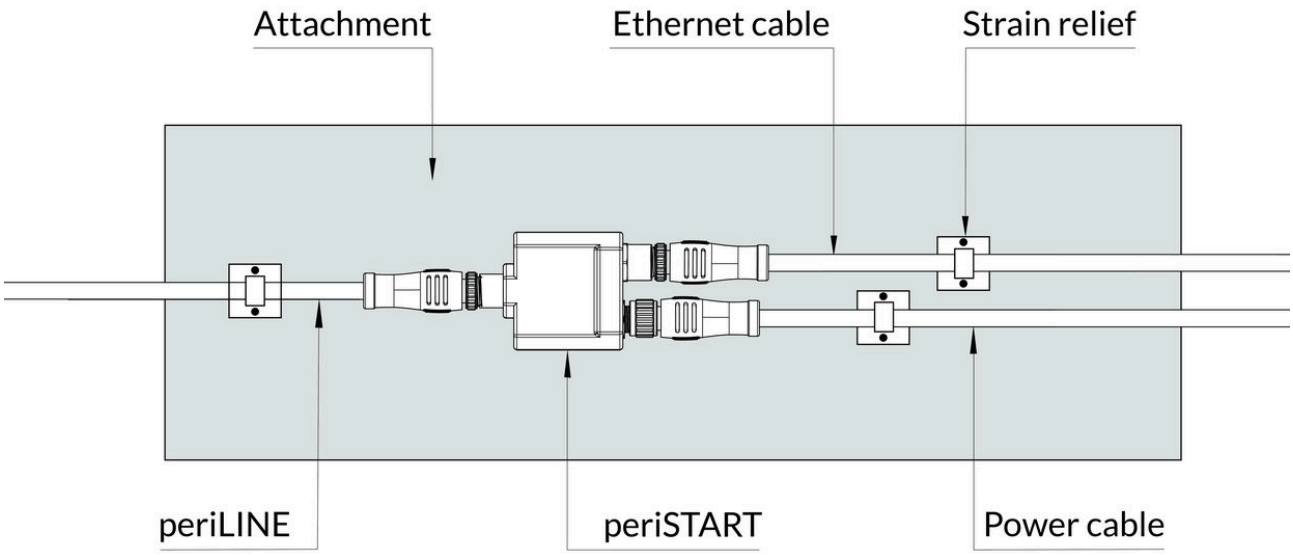

Figure 34: periSTART installation

periSTART standard, periNODE GPIO, and periSWITCH 3-port are designed to be supported solely by their cables. To ensure stability and prevent any strain, the cables should be securely fastened near the device using a strain relief to a solid attachment.

Note: The Perinet Smart Component *periSTART standard* is ideally used near a network outlet or a standard ethernet switch. To maintain optimal performance, it's recommended that both the power and Ethernet cables should not exceed a length of 3 meters.

Note: A periNODE should be connected to its sensor or actuator with a cable no longer than 3 meters.

Note: Each periNODE variant, periSWITCH, and periSTART must be connected to other network components with a periLINE of less than 15 meters in length. This limit is applicable for cables with unshielded data lines; for shielded data lines, the maximum length is 40 meters.

For personal safety and to prevent property damage, please also refer to our ***Safety Instructions***.

7 Labeling and Ordering Information

7.1 Product Marking

Product markings provide essential information about the device. This same information is also accessible in a machine-readable format, detailed in the *Data Matrix Code* section.

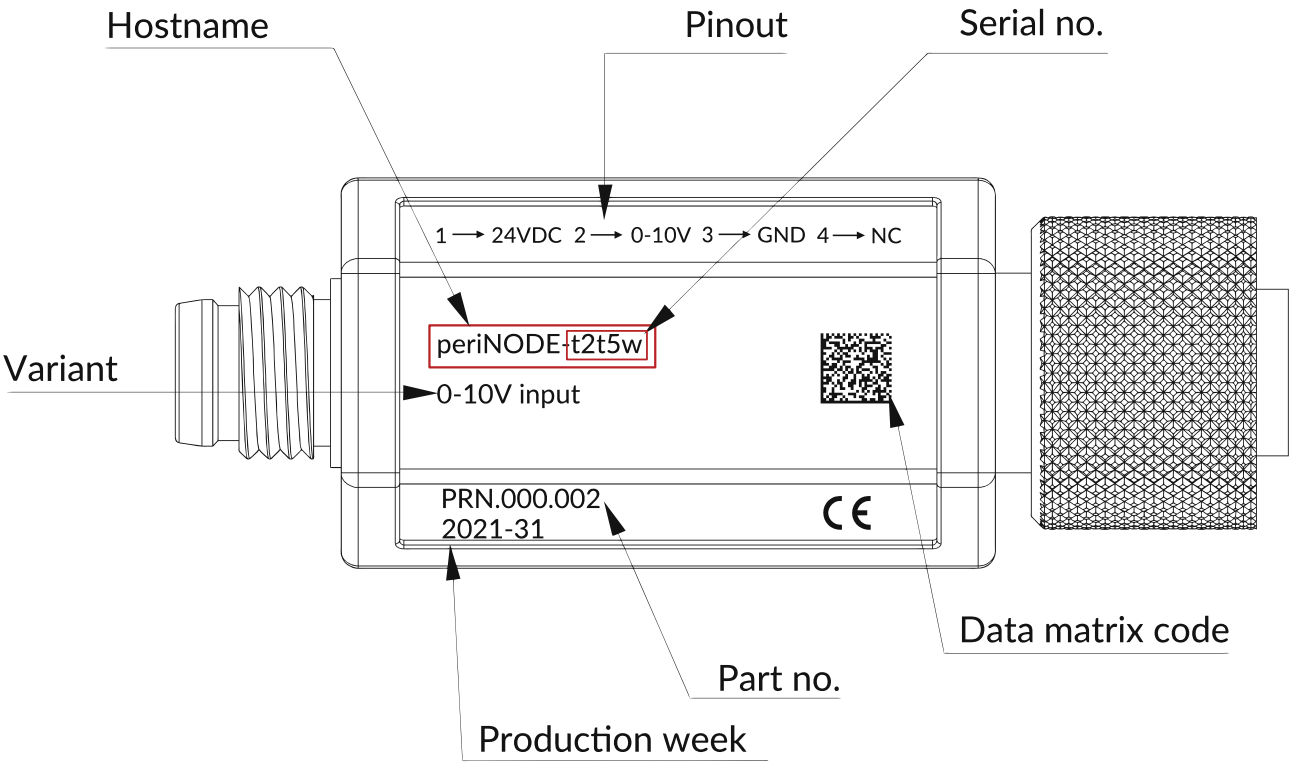

Figure 35: Product marking

- **Pinout**: Details for sensor or actuator interface on periNODE.
- **Hostname**: The unique hostname of the Perinet Smart Component.
- **URL**: Access periNODE in your Local Area Network by typing “https://” followed by its hostname and “.local/”. For instance, “periNODE-t2t5w” translates to `"https://periNODE-t2t5w.l`
- **Serial no.**: The last five characters of the hostname, serving as the device’s unique serial number. For example, “periNODE-t2t5w” has the serial number “t2t5w”.
- **Variant**: Indicates the interface signal type for periNODE, specifying the compatible sensor or actuator interface. For instance, “0-10V input” means it can connect to sensors with a 0-10V output.
- **Part no.**: The Manufacturer Part Number, serving as a specific identifier for the device type in the market.

- **Production week**: The production date of the device, formatted as YYYY-CW. “2020-36” denotes production in the 36th week of 2020.
- **CE sign**: Denotes CE compliance of the product (refer to *Qualifications and Approvals*).
- *Data Matrix Code*: A machine-readable code containing the device’s detailed information.

7.2 Data Matrix Code

To facilitate machine readability, key information about Perinet Smart Components is encoded in a two-dimensional Data Matrix (ECC200) code. This code can be read using an optical reader, such as an industrial scanner or a smartphone equipped with a suitable app.

The encoded information is structured as follows:

- Line 1: URL, which includes the hostname, which includes the serial number.
- Line 2: Manufacturer Part Number.
- Line 3: Production calendar week.

7.3 Ordering Information

Part number	Product Name
PRN.000.002	periNODE 0-10V
PRN.000.006	periNODE Pt100
PRN.000.010	periNODE GPIO
PRN.000.003	periSTART standard
PRN.000.008	periSWITCH 3-port
PRN.000.017	periLINE 0.2m
PRN.000.009	periLINE 1m
PRN.000.011	periLINE 2m
PRN.000.013	periLINE 5m
PRN.000.015	periLINE 10m

Table 31: Ordering information

7.4 Packaging

Perinet Smart Components are shipped in packaging tailored to each product type. For all periNODE variants, the boxes contain 50 devices each. The periSWITCH 3-port and periSTART standard models are packaged in boxes of 25 devices each.

8 Contact & Support

For customer support, please call us at **+49 30 863 206 701** or send an e-mail to *support@perinet.io*.

For complete contact information visit us at www.perinet.io

9 Safety Instructions

This section contains information that you must observe for your personal safety and to avoid damage of property. The instructions for your personal safety are highlighted by a warning triangle, while information on damage of property alone is given without a warning triangle. Depending on the hazard level, the warnings are shown in decreasing order as follows.

⚠ DANGER	means that death or serious injury will occur if the appropriate precautions are not taken.
⚠ WARNING	means that death or serious injury may occur if the appropriate precautions are not taken.
⚠ CAUTION	with warning triangle, means that minor injury may occur if the appropriate precautions are not taken.
CAUTION	without warning triangle, means that property damage may occur if the appropriate precautions are not taken.
ATTENTION	means that an undesirable result or condition may occur if the corresponding precaution is not taken.
INFO	information on needed measures to guard against system failures.

Table 32: Safety instructions

If several hazard levels occur, the warning notice for the highest level is always used.
If the warning triangle is used in a warning against personal injury, then a warning against property damage can also be added to the same warning.

9.1 Power Supply

⚠ DANGER

Connection of the 24 V device only via safety extra-low voltage/protective extra-low voltage

The device is designed for operation with a directly connectable safety extra-low voltage (Safety Extra-Low Voltage, SELV) or Protective Extra-Low Voltage (PELV).

Failure to observe this warning may result in electric shock or damage to property.

Therefore, only safety extra-low voltages (SELV) according to IEC 60950-1 / EN 60950-1 / VDE 0805-1 or protective extra-low voltages (PELV) according to EN 50178 may be connected to the supply connections.

ATTENTION

Connecting the 24 V device to a safety extra-low voltage (SELV) turns it into a protective extra-low voltage (PELV). The connection point between ground and PE is established by the device according to the IEC 61851 standard.

9.2 Protection

CAUTION

Property damage

Property damage caused by electrostatic charging.

- Follow the measures necessary for the handling of components prone to electrostatic charging.

⚠ WARNING

Unwanted heat generation or fire due to insufficient fuse protection

The internal fuses of the device are designed for device protection only. The system installer or operator is responsible for the necessary line protection.

9.3 Repair

Repairs are not permitted. Defective devices must be disposed of in accordance with environmental requirements.

CAUTION

Dangers due to unauthorized opening of the device

Unauthorized opening of the device can cause considerable damage to property or danger to the user.

CAUTION

Voiding of the manufacturer's warranty due to unauthorized modifications to the device

Modifications to the equipment are not permitted. Failure to comply will invalidate the manufacturer's warranty expires.

Only put undamaged components in operation.

INFO

Perinet expressly notes that devices to which this document pertains are unsuitable for use as part of or in connection with medical implants or as important components of life support systems, where failure could lead to severe injuries to people. The components used and the proven level of reliability don't meet the necessary requirements for such applications. In order to avoid damage to devices and equipment as well as injuries to or death of people, the user or application developer must implement suitable, well thought out measures to guard against system failures.

The rights of third parties may be violated through use in the transport market of devices which are described in this document. Perinet can provide no warranty whatsoever that the rights of third parties aren't violated through use of the devices. If you are planning on this sort of use, please contact Perinet in order to clarify any potential questions regarding patent or property rights.

9.4 Qualification of the Personnel

The devices may only be mounted, installed, commissioned and maintained by persons who are familiar with the handling of such devices and who are qualified for the described activities.

This qualification includes:

- Training regarding the installation, connection, earthing, commissioning and maintenance of electrical devices (EN 50 110-1/-2 / VDE 0105 part 100).
- Training regarding the current standards of electrical engineering and safety technology.
- First aid training.

A Smart Components API Specification

```
//API_VERSION=11


syntax="proto3";

message FwUpdate {
    message Header {
        uint32 length = 1;
        uint32 chksum = 2;
    }
    string signature = 1; // signature
    Header header = 2; // meta information
    bytes data = 3; // firmware update binary data
}syntax="proto3";

package perinet.periNODE.api;

/**
*   Message that represents a measured typed sample. see also NodeType
*   Example JSON for a APPTYPE=DISTANCE_SOURCE {"incarnation" : 1,"sequence_number":100,"interface_type":"DISTANCE_SOURCE", "data": [ {"unit":"mm","distance": 180 } ] }
*   Example JSON for a APPTYPE=PH_SOURCE {"incarnation" : 1,"sequence_number":100,"interface_type":"PH_SOURCE", "data": [ {"unit":"pH","ph": 0.1},{"unit":"°C","temp": 22.6 } ] }
*   Example JSON for a APPTYPE=DIGITAL_IO_SOURCE {"incarnation" : 1,"sequence_number":100,"interface_type":"DIGITAL_IO_SOURCE", "data": [ {"unit":"","input":{"port":"photoelectric",value:1}} ] }
*/
message Sample {
    message InterfaceSample {
        string unit = 1; // base units according to chosen APPType for the source interface might be Si based, empty if n.a.
        oneof sample_values {
            uint32 raw = 2; // value for Interfaces configured as RawType
            float temp = 3; // value for Interface sources of type TEMPERATURE_SOURCE
            float pH = 4; // value for Interface sources of type PH_SOURCE
            uint32 distance = 5; // value for Interface sources of type DISTANCE_SOURCE
            IOStatus io = 6; // value for Interface sources of type DIGITAL_IO_SOURCE
            Blind_Control blind_ctrl = 7; // value for Interface sources of type BLIND_CONTROL_SOURCE
            float analog = 8;
            float acceleration = 9;
        }
    }

    /*required*/ uint32 incarnation = 1; // partial unique number of a message sample, which becomes unique in combination with sequence_number
```

```
    /*required*/ uint64 sequence_number = 2; // partial unique number of a message sample, which becomes unique in combination with incarnation
    repeated InterfaceSample data = 4; // a collection of samples, which are defined by the InterfaceType.
}

enum PortNumber {
    Unassigned = 0;
    PORT1 = 1;
    PORT2 = 2;
}

/**
* sample type for NodeType of type DIGITAL_IO_SOURCE
*/
message IOStatus {
    PortNumber port = 1;
    bool value = 2;
}

enum Blind_Control {
    STOP = 0;
    UP = 1;
    DOWN = 2;
}
syntax="proto3";

// package perinet;
package perinet.periNODE.api;

message Certificate {
    bytes certificate = 1;
}

message CertPKBundle {
    string cert_pk_bundle = 1;
}

message PrivateKey {
    bytes private_key = 1;
}

message SecurityConfig {
    bool enable_user_role = 1;
    string root_cert = 2; //optional
    string host_cert = 3; //optional
    string client_cert = 4; //optional
}

message FwUpdateStream {
    message Header {
        uint32 length = 1; //byte count
        uint32 digest = 2; //sha256 digest over data
    }
    string signature = 1; //signed header over sha256
    Header header = 2;
```

```
    bytes data = 3; // firmware update binary data
}
/*
//header
{
    "length":286720, //cat _build/update_periCORE.img|wc -c
    "digest":"U0hBMjU2KF9idWlsZC91cGRhdGVfcGVyaUNPUkUuaW1nKT0gYWI1NjFiOGM2NmJjYTdmNDdhM2E3ZGFhNTl=" //openssl dgst -sha256  _build/update_periCORE.img | base64 -w0
}
*/syntax = "proto3";

package perinet.periNODE.api;

message MqttBrokerList {
  repeated string broker = 1;
}

syntax="proto3";

import public "google/protobuf/empty.proto";

import "google/api/annotations.proto"; //needs the https://github.com/googleapis/googleapis repo as include path

import "Sample.proto";
import "NodeConfig.proto";
import "NodeInfo.proto";
import "MqttBrokerList.proto";
import "SecurityConfig.proto";
import "FwUpdate.proto";

package perinet.periNODE.api;

//needs the https://github.com/googleapis/googleapis repo as include path

/**
*   http REST compliant service of a periNODE.
*/
service PeriNodeService {
    // request general purpose information on the Node
    rpc GetInfo (google.protobuf.Empty) returns (NodeInfo){
        option (google.api.http) = {
            get: "/info"
        };
    };
    // request the configuration information of a periNODE
    rpc GetConfiguration (google.protobuf.Empty) returns ( NodeConfig ){
        option (google.api.http) = {
            get: "/config"
        };
    };
    // request mqtt broker list information of a periNODE
    rpc GetMqttBrokerList (google.protobuf.Empty) returns ( MqttBrokerList ){
        option (google.api.http) = {
```

```
            get: "/mqtt"
        };
    };
    // update the configuration of a periNODE via PATCH or url encoded GET request
    rpc SetConfiguration (NodeConfig) returns (NodeConfig){
        option (google.api.http) = {
            patch: "/config"
            body: "*"
        };
    };

    // request the last sensor information of a periNODE
    rpc GetSample (google.protobuf.Empty) returns ( Sample ){
        option (google.api.http) = {
            get: "/sample"
        };
    };

    // provide the actuator data for a periNODE
    rpc SetSample (Sample) returns ( Sample ){
        option (google.api.http) = {
            patch: "/sample"
            body: "*"
        };
    };

    // request the security configuration of a periNODE
    rpc GetSecurityConfig (google.protobuf.Empty) returns ( SecurityConfig ){
        option (google.api.http) = {
            get: "/security"
        };
    };
    // request the client certificate for a periNODE
    rpc GetClientCert (google.protobuf.Empty) returns ( Certificate ){
        option (google.api.http) = {
            get: "/security/client-cert"
        };
    };
    // request Perinet root certificate for a periNODE
    rpc GetRootCert (google.protobuf.Empty) returns ( Certificate ){
        option (google.api.http) = {
            get: "/security/root-cert"
        };
    };
    // request Perinet root certificate for a periNODE
    rpc GetHostCert (google.protobuf.Empty) returns ( Certificate ){
        option (google.api.http) = {
            get: "/security/host-cert"
        };
    };
    // provide security configuration for a periNODE
    rpc SetSecurityConfig (SecurityConfig) returns ( google.protobuf.Empty ){
        option (google.api.http) = {
```

```
199             patch: "/security"
200             body: "*"
201         };
202     };
203     // provide client certificate including private key for a periNODE
204     rpc SetClientCertBundle (CertPKBundle) returns ( google.protobuf.↩
    Empty ){
205         option (google.api.http) = {
206             patch: "/security/client-cert"
207             body: "*"
208         };
209     };
210     // provide user defined root certificate for a periNODE
211     rpc SetRootCert (Certificate) returns ( google.protobuf.Empty ){
212         option (google.api.http) = {
213             patch: "/security/root-cert"
214             body: "*"
215         };
216     };
217     // provide user defined certificate and private key for a periNODE
218     rpc SetHostCertBundle (CertPKBundle) returns ( google.protobuf.Empty ↩
    ){
219         option (google.api.http) = {
220             patch: "/security/host-cert"
221             body: "*"
222         };
223     };
224     // update periFIRMWARE, does not affecting configuration nor ↩
    manufacturing data
225     rpc UpdateFirmware (stream FwUpdate) returns ( google.protobuf.Empty ↩
    ){
226         option (google.api.http) = {
227             put: "/update"
228             body: "*"
229         };
230     };
231     // invoke softreset of a periNODE
232     rpc SetReboot (google.protobuf.Empty) returns (google.protobuf.Empty)↩
     {
233         option (google.api.http) = {
234             patch: "/reboot"
235             body: "*"
236         };
237     }
238     // reset non-volatile configuration settings to its default values.
239     rpc SetConfigReset (google.protobuf.Empty) returns (google.protobuf.↩
    Empty) {
240         option (google.api.http) = {
241             patch: "/config/reset"
242             body: "*"
243         };
244     };
245     // reset security configuration to production state, resets root_cert↩
    , host_cert and client_cert.
246     rpc SetSecurityReset (google.protobuf.Empty) returns (google.protobuf↩
    .Empty) {
```

```
247         option (google.api.http) = {
248             patch: "/security/reset"
249             body: "*"
250         };
251     };
252     // reset periNODE into factory defaults.
253     rpc FactoryReset (google.protobuf.Empty) returns (google.protobuf.↩
    Empty) {
254         option (google.api.http) = {
255             patch: "/reset"
256             body: "*"
257         };
258     };
259 
260     // mqtt message stream towards the sensor adapter periNODE, which ↩
    acts then as an actuator or data sink, respectively.
261     rpc Publish (stream Sample) returns (google.protobuf.Empty);
262     // mqtt message stream from the sensor adapter periNODE, which acts ↩
    as an data source.
263     rpc Subscribe (google.protobuf.Empty) returns (stream Sample);
264 }
265 syntax="proto3";
266 
267 // package perinet;
268 package perinet.periNODE.api;
269 
270 // representation of the available applications, implementations of ↩
    sensors respectively.
271 enum InterfaceType {
272     INVALID_APPTYPE = 0; //representation of an invalid sensor interface ↩
    value
273     RAW_SOURCE = 1; // sensor interface of any raw value, debug purpose
274     TEMPERATURE_SOURCE = 2; // sensor interface of a temperature sensor
275     PH_SOURCE = 3; // sensor interface of a ph sensor
276     DISTANCE_SOURCE = 4; // sensor interface of a distance sensor
277     DIGITAL_IO_SOURCE = 5; // sensor interface of a digital io input
278     BLIND_CONTROL_SOURCE = 6; // sensor interface of a switch control ↩
    input
279     DIGITAL_IO_SINK = 7; // sensor interface of a digital io output
280     BLIND_CONTROL_SINK = 8; // sensor interface of a discrete output to ↩
    control shutter or blinds {UP,DOWN,STOP} value range
281     ANALOG_OUT_SINK = 9; // sensor interface of a analoguous output ↩
    0.0..1.0 value range
282     ACCELERATION_SOURCE = 10; // sensor interface of acceleration sensor
283 }
284 
285 message NodeConfig {
286     message Interface {
287         InterfaceType type = 1; //the type of interface, the particular ↩
    interface has been configured to. Implementation specific.
288         string element_name = 2; // element name of a particular ↩
    interface, like an sensor source or an actuator sink.
289         float period_seconds = 3; // in seconds, 0 means only event based↩
     (triggered) publishing
290         uint32 samples_per_period = 4;  //defines how many values shall ↩
    be sampled within a period,
```

```
                                        //the published value will be the rounded average
        // repeated Trigger trigger = 5;
    }
    string application_name = 1; //

    string mqtt_broker_name = 2; //

    repeated Interface configs = 3; //

    // InterfaceConfig sink_interface_config = 4; // first interface
    // InterfaceConfig source_interface_config = 5; // second interface
}
syntax="proto3";

package perinet.periNODE.api;

message SwVersion {
    uint32 api = 1; // API compatibility incarnation

    uint64 build = 2; // build iteration

    uint32 version_number = 3; // periFIRMWARE feature level incarnation
}

message VersionInfo {
    SwVersion firmware_version = 1; // periFIRMWARE version information
    string firmware_variant = 4; // product variant, the periFIRMWARE has been aligned with. e.g. periNODE 0-10V; periSTART; periSWITCH 3-port
}

//changes to that node might need to be changed to NodeTXTInfo
message NodeInfo {
    VersionInfo version_info = 1; // firmware version information
    string manufacturer = 2; // manufacturer identifier
    string hostname = 3; // host network identifcation name, e.g. periNODE-<id>.local
    string mac_address = 4; // unique mac address
    string product_charge = 5; // production batch identifier
    string product_part_number = 6; // product part identifier
    string product_serial = 7; // serial number of the product
    string product_name = 8; // calling name of the product
    string product_version = 9; // version of the product at production time
    string pericore_charge = 10; // batch identifier of the included periCORE
    string pericore_part_number = 11; // periCORE part identifier
    string pericore_serial = 12; // periCORE serial number
    string pericore_version = 13; // periCORE version identifier
}
```

B List of Figures

C List of Listings

D List of Tables

E Glossary

API Application Programming Interface. 36, 47

DNS-SD DNS Service Discovery [1] is a way of using standard DNS programming interfaces, servers and packet formats to browse the network for services. 55

GPIO General-Purpose Input/Output. 49

HTTP Hypertext Transfer Protocol is an application-layer protocol for transmitting hypermedia documents, such as HTML. 46

IIoT Industrial Internet of Things. 1

IoT Internet of Things. 41, 42

JSON JavaScript Object Notation is standard text-based format for representing structured data based on JavaScript object syntax. 45–47, 49, 50, 55

LLMNR The Link-Local Multicast Name Resolution is a protocol based on the Domain Name System packet format that allows IPv6 hosts to perform name resolution for hosts on the same local link. 36

mDNS multicast Domain Name Service [2], a protocol that implements a local distributed name resolving mechanism. 36, 45, 55

MQTT Message Queuing Telemetry Transport is a lightweight, publish-subscribe based network protocol that transports messages between devices. 41–43, 45, 47, 55, 56

mTLS Mutual TLS extends the TLS protocol by requiring clients to pass certificates, allowing to provide authorization mechanisms of Application services. 56

REST REpresentational State Transfer, a web API style. 36, 39, 40, 45, 47, 49, 50, 55, 58–60

SPE Single Pair Ethernet. 1

TLS Transport Layer Security Protocol. Used by Application Protocols like MQTT or HTTP to allow secure data transfer. 36, 55

F References

[1] S. Cheshire and M. Krochmal. *DNS-Based Service Discovery*. RFC 6763. http://www.rfc-editor.org/rfc/rfc6763.txt. RFC Editor, Feb. 2013. URL: http://www.rfc-editor.org/rfc/rfc6763.txt.

[2] S. Cheshire and M. Krochmal. *Multicast DNS*. RFC 6762. http://www.rfc-editor.org/rfc/rfc6762.txt. RFC Editor, Feb. 2013. URL: http://www.rfc-editor.org/rfc/rfc6762.txt.

[3] Perinet GmbH. Perinets IoT Application Concepts Application Note. PRN.100.552. https://docs.perinet.io/.

[4] Perinet GmbH. Security Certificates Installation Guide. PRN.100.447. https://docs.perinet.io/PRN100447-SecurityCertificatesInstallationGuide.pdf.

Revision History

Revision	Date	Author(s)	Description
1.0	September 14, 2021	Soeren Hoeckner	Initial release
2	July 12, 2022	Perinet, asch, show, tkla, msen, kwal	Corrected code listing in Section 3; Correct specifications in Section 2; Update Format; Update referenced Documents
3	2024-02-08	clip	Editorial review, changed RBAC roles from "reader" to "user"